DS4LAW

Workshop 1: Algorithms (foundations)

Kody Moodley, PhD.

The Faculty of Law & The Institute of Data Science @ Maastricht University

Maastricht University

Institute of Data Science

A little about me

- Born and bred in South Africa
- Trained as a Computer Scientist
- BSc & MSc in Pretoria
- PhD in Manchester
- Industry - **creating algorithms** (mobile applications)
- 1st Postdoc in Maastricht

CSIR

MANCHESTER 1824
The University of Manchester

Maastricht University

“In theory, there is no difference between theory and practice, in practice, there is.” - Jan L. A. Van De Snepscheut (Caltech)

A little about you?

- Brief introduction (name & affiliation)
- What is your interest: DS4LAW or LAW4DS?
- Experience with algorithms
- What you hope to gain from the session

Aims

- Basic literacy of algorithms in the context of Computer Science
- LAW4DS:
    - Understanding mechanisms of algorithms should aid literacy in formulating questions, and focusing discussions, about their regulation.
    - Intuition for distinguishing between true algorithmic intelligence vs. apparent algorithmic intelligence
- DS4LAW:
    - Theoretical foundations of algorithms & role in computer science
    - Basic practical experience with creating algorithms

Roadmap

Mostly talking

- Part 1A: Algorithms are all around us...but what exactly are they?
- Part 1B: Algorithms in Computer Science
- Part 2: Examples of simple algorithms
- Part 3: Examples of algorithms that are ubiquitous in contemporary software
- Part 4: Ethical significance of algorithms

Mostly doing

- Python programming in Jupyter notebooks
- Write and test some algorithms!

Roadmap

- Part 1: Algorithms are all around us...but what are they?
  - Examples in the real world (well known)
  - Examples in law
  - Definition outside of CS
- Part 2: Algorithms in Computer Science
  - Instructions need to be specified in some language
  - What languages can computers understand?
  - Spectrum of programming languages & the role of compilers
- Part 3: Examples of simple algorithms
  - Sorting & find highest value (Link to Excel)
- Part 4: Examples of ubiquitously reused algorithms
- Part 5: Ethical significance of algorithms

Part 1A

Algorithms are all around us...but what are they?

Definition for Algorithm

“a set of rules that must be followed when solving a particular problem.”

- Oxford Learner’s Dictionary

“A set of specific, step-by-step instructions for taking an input and converting [it] into an output.”

- John Danaher, The Philosophical Importance of Algorithms, 2015.

In short: “a set of step-by-step instructions for executing a task.”

Examples

Making a peanut butter & jelly sandwich

1. Get a piece of bread
2. Spread peanut butter on it
3. Get another piece of bread
4. Spread jelly on it
5. Put the two pieces of bread together

Who is issuing the instructions?
Who is executing the instructions?

What are the inputs & outputs of the algorithm?

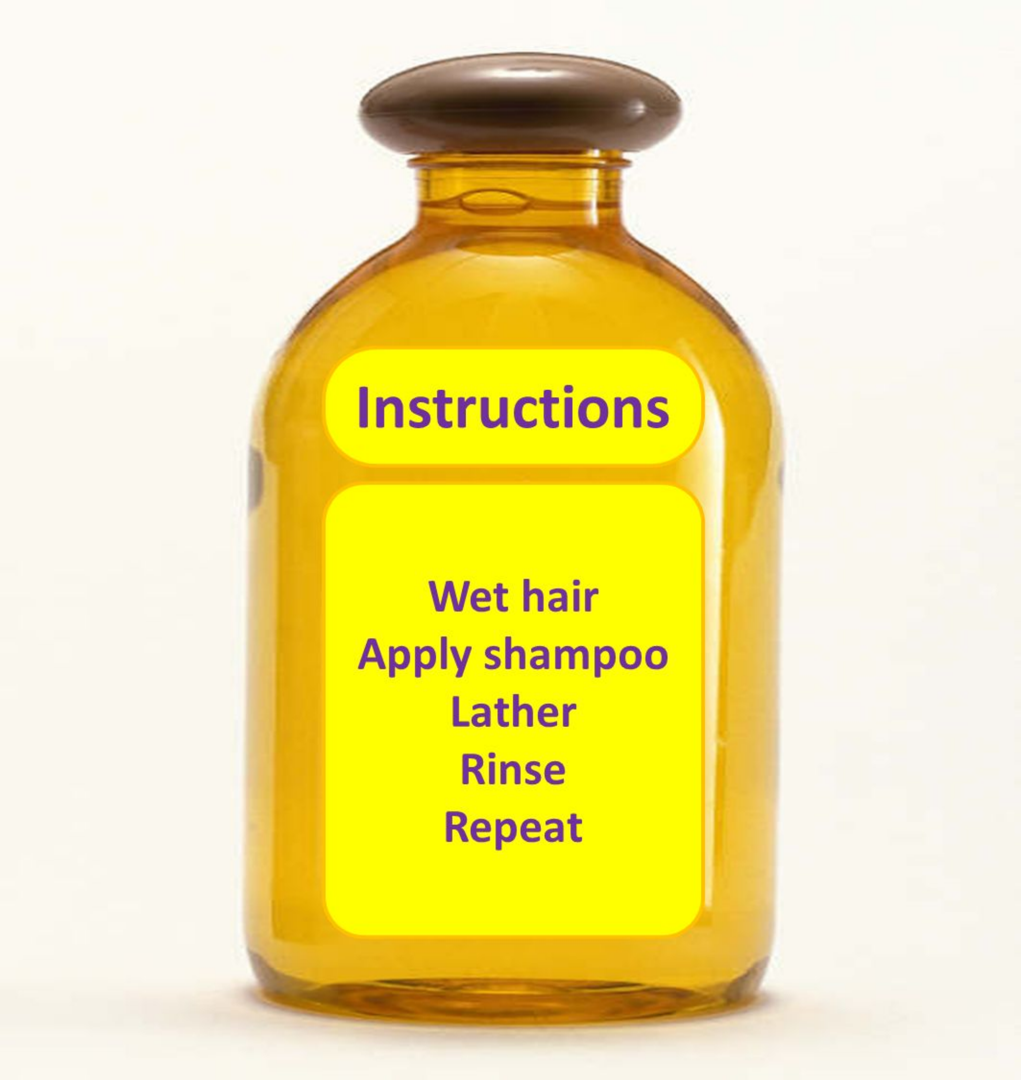
Examples

Shampooing hair

Anything strange about this algorithm?

Who is issuing the instructions?
Who is executing the instructions?

What are the inputs & outputs of the algorithm?

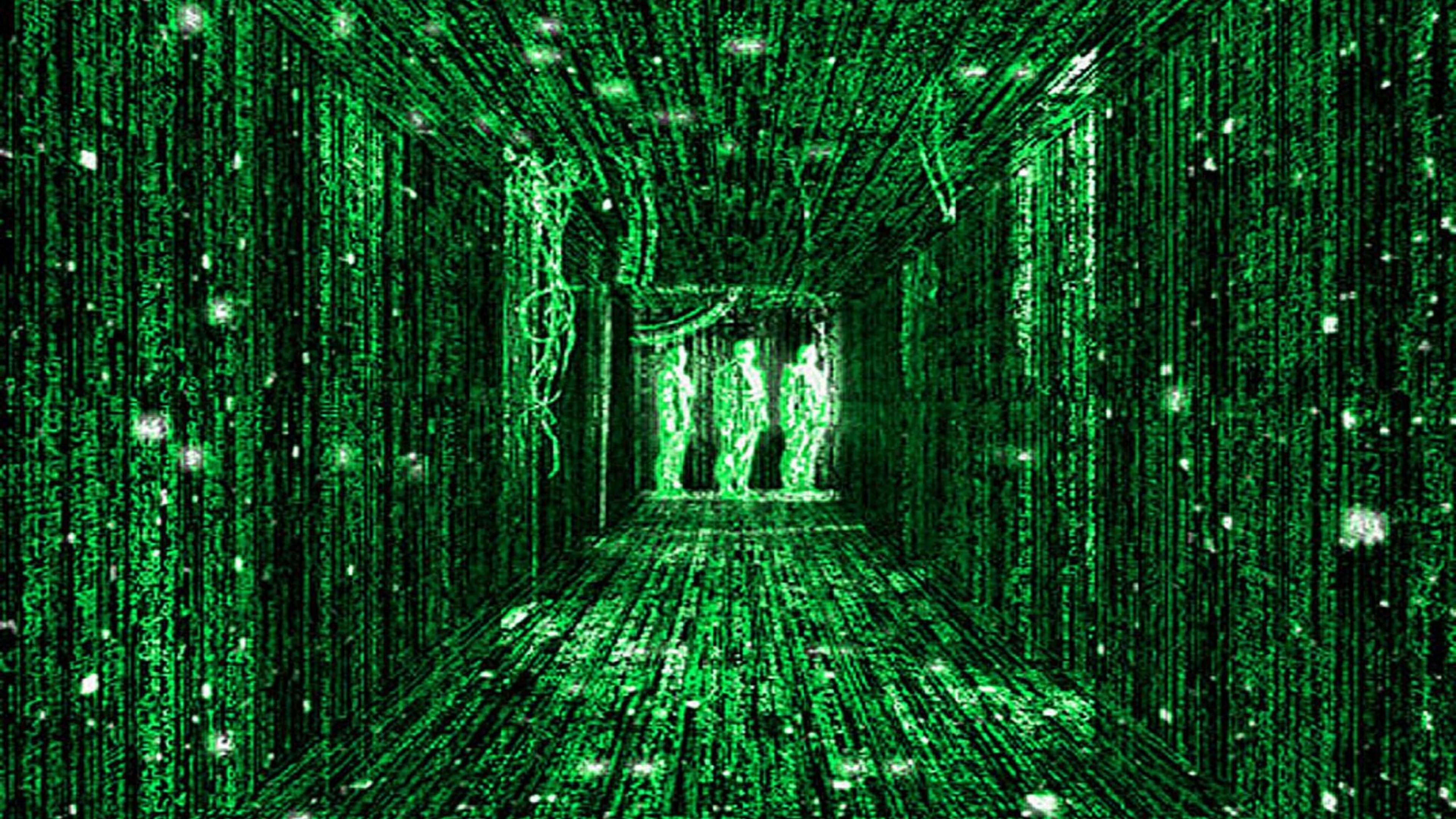


Instructions

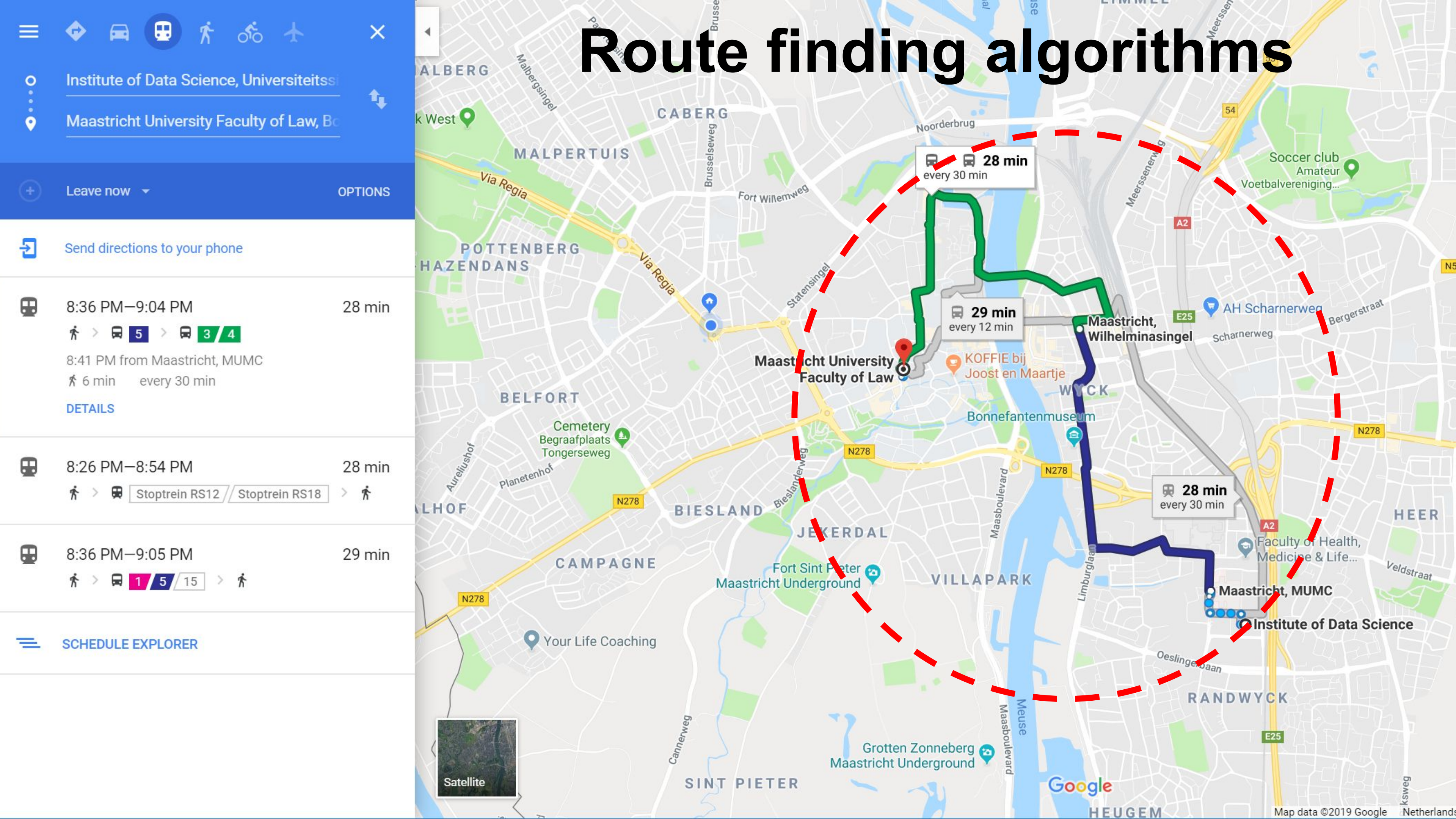
Wet hair
Apply shampoo
Lather
Rinse
Repeat

Part 1B

Algorithms in Computer Science



Route finding algorithms



Institute of Data Science, Universiteitss

Maastricht University Faculty of Law, Bo

Leave now ▾ OPTIONS

Send directions to your phone

8:36 PM—9:04 PM 28 min

5 > 3 / 4

8:41 PM from Maastricht, MUMC

6 min every 30 min

DETAILS

8:26 PM—8:54 PM 28 min

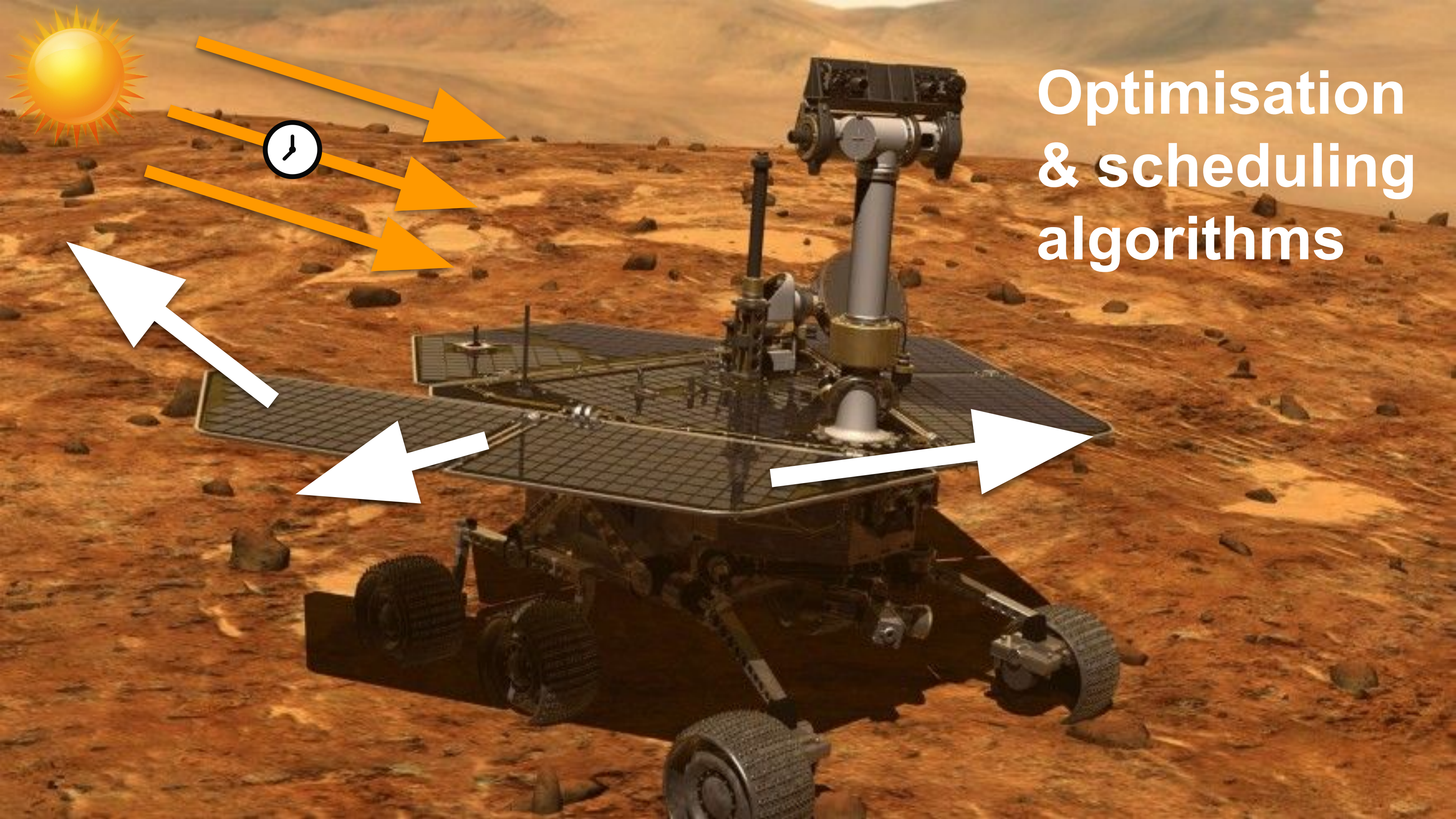
Stoptrein RS12 / Stoptrein RS18

8:36 PM—9:05 PM 29 min

1 / 5 / 15

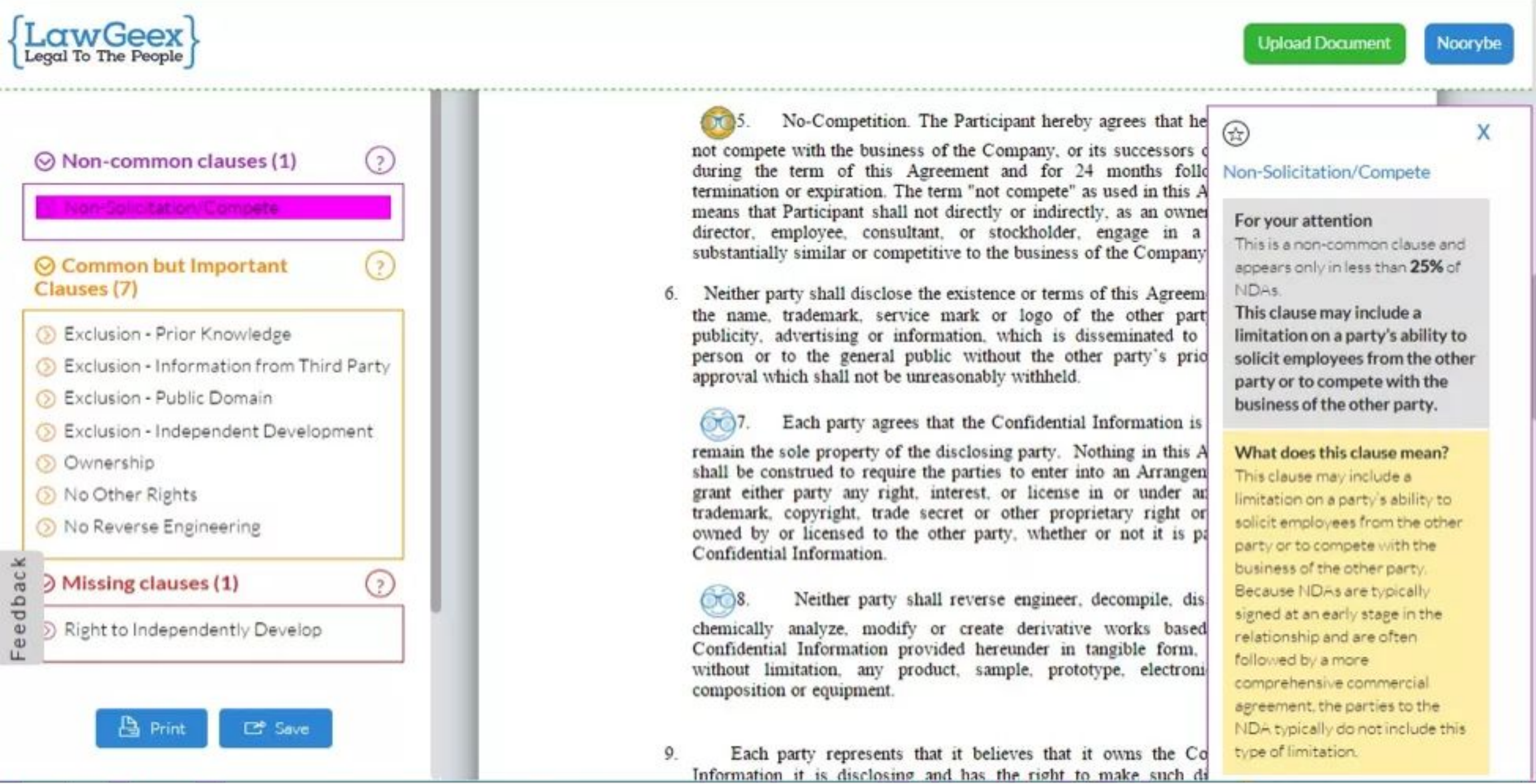
SCHEDULE EXPLORER

Audio & Video compression algorithms



Optimisation & scheduling algorithms

Contract review using Machine Learning algorithms



Non-common clauses (1)

Non-Solicitation/Compete

Common but Important Clauses (7)

- Exclusion - Prior Knowledge
- Exclusion - Information from Third Party
- Exclusion - Public Domain
- Exclusion - Independent Development
- Ownership
- No Other Rights
- No Reverse Engineering

Missing clauses (1)

- Right to Independently Develop

5. No-Competition. The Participant hereby agrees that he not compete with the business of the Company, or its successors during the term of this Agreement and for 24 months follo termination or expiration. The term "not compete" as used in this A means that Participant shall not directly or indirectly, as an owne director, employee, consultant, or stockholder, engage in a substantially similar or competitive to the business of the Company

6. Neither party shall disclose the existence or terms of this Agreem the name, trademark, service mark or logo of the other part publicity, advertising or information, which is disseminated to person or to the general public without the other party's prio approval which shall not be unreasonably withheld.

7. Each party agrees that the Confidential Information is remain the sole property of the disclosing party. Nothing in this A shall be construed to require the parties to enter into an Arrangem grant either party any right, interest, or license in or under an trademark, copyright, trade secret or other proprietary right or owned by or licensed to the other party, whether or not it is pa Confidential Information.

8. Neither party shall reverse engineer, decompile, dis chemically analyze, modify or create derivative works based Confidential Information provided hereunder in tangible form, without limitation, any product, sample, prototype, electroni composition or equipment.

9. Each party represents that it believes that it owns the Co Information it is disclosing and has the right to make such di

Non-Solicitation/Compete

For your attention

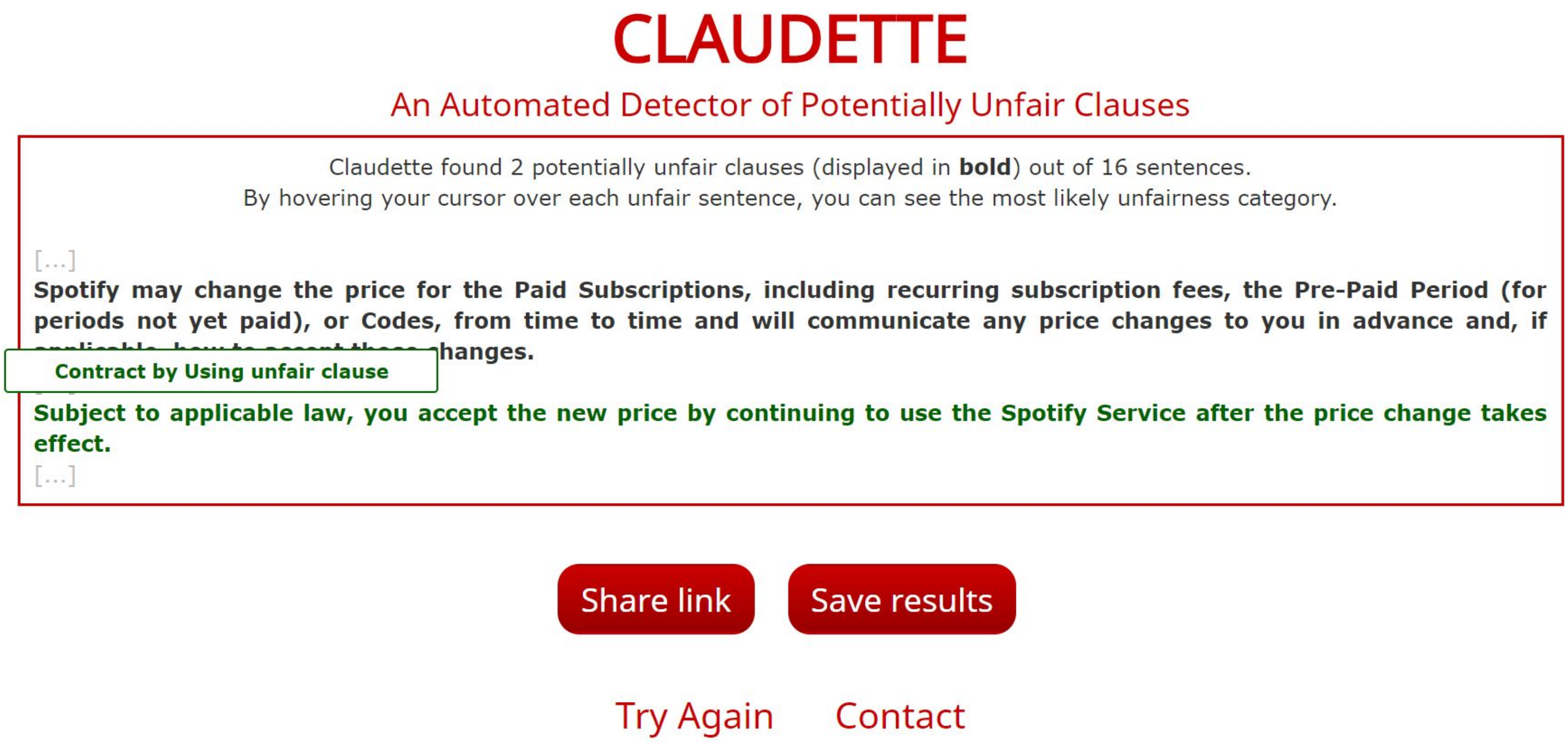
This is a non-common clause and appears only in less than **25%** of NDAs.

This clause may include a limitation on a party's ability to solicit employees from the other party or to compete with the business of the other party.

What does this clause mean?

This clause may include a limitation on a party's ability to solicit employees from the other party or to compete with the business of the other party. Because NDAs are typically signed at an early stage in the relationship and are often followed by a more comprehensive commercial agreement, the parties to the NDA typically do not include this type of limitation.

Unfair clause detection using Machine Learning



CLAUDETTE

An Automated Detector of Potentially Unfair Clauses

Claudette found 2 potentially unfair clauses (displayed in **bold**) out of 16 sentences.
By hovering your cursor over each unfair sentence, you can see the most likely unfairness category.

[...]

Spotify may change the price for the Paid Subscriptions, including recurring subscription fees, the Pre-Paid Period (for periods not yet paid), or Codes, from time to time and will communicate any price changes to you in advance and, if applicable, how to accept these changes.

Contract by Using unfair clause

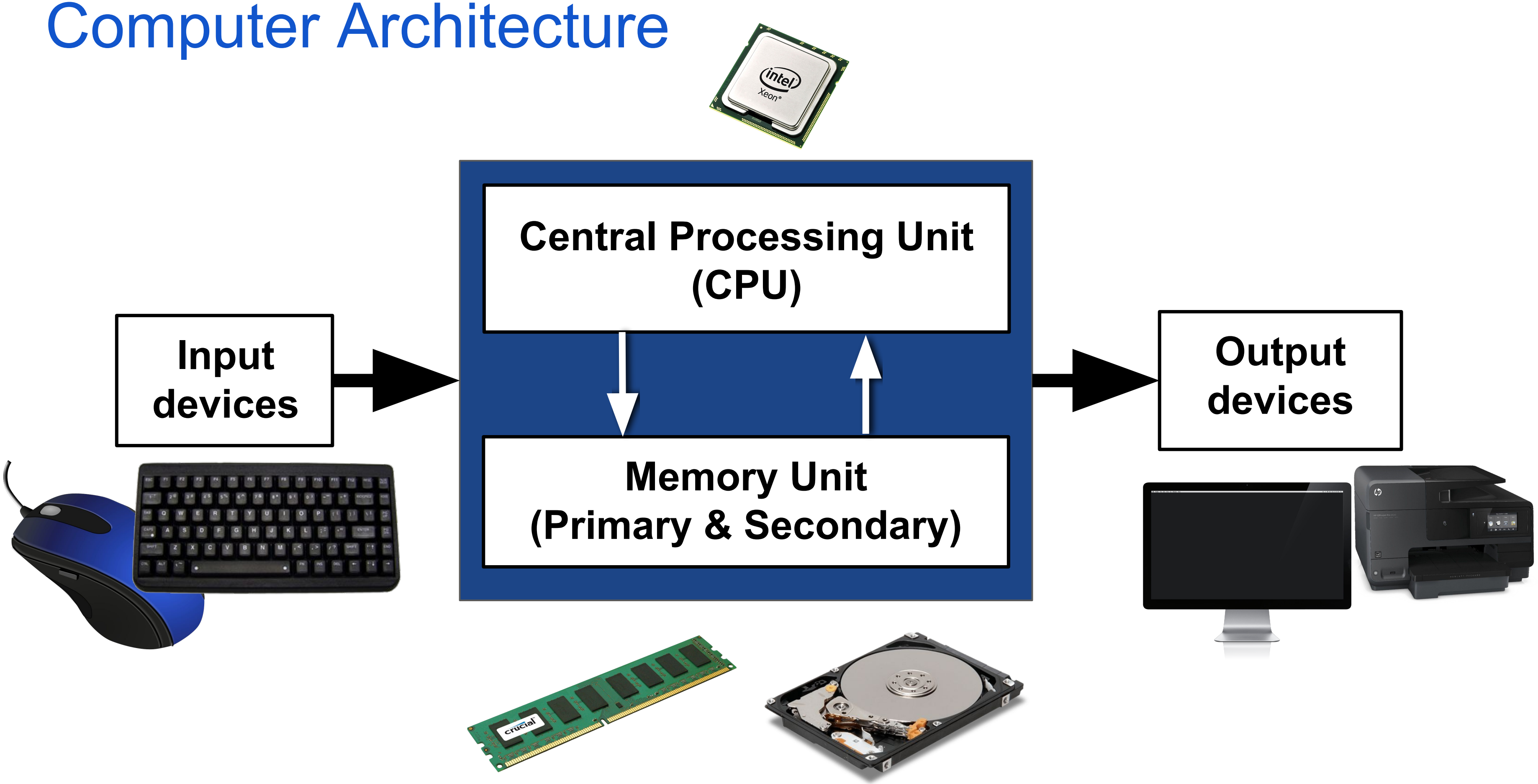
Subject to applicable law, you accept the new price by continuing to use the Spotify Service after the price change takes effect.

[...]

Algorithms in Computer Science

Who is issuing the instructions? **Humans**
Who is executing the instructions? **Computers**

Computer Architecture



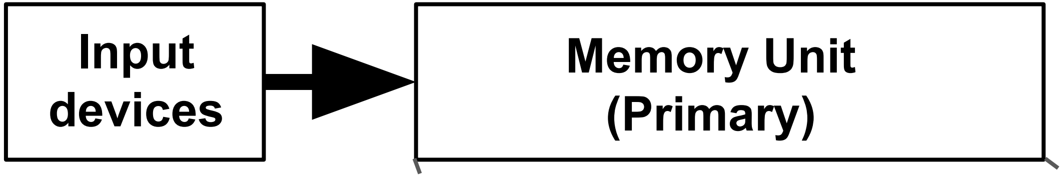
Input devices

Central Processing Unit (CPU)

Memory Unit (Primary & Secondary)

Output devices

Computer Architecture



23 + 34?

STORAGE

1	2	3
Value: “23”	**Instruction:** “add(1,3)”	**Value:** “34”
...	...	...

How many bits does a computer with 4GB of RAM have?

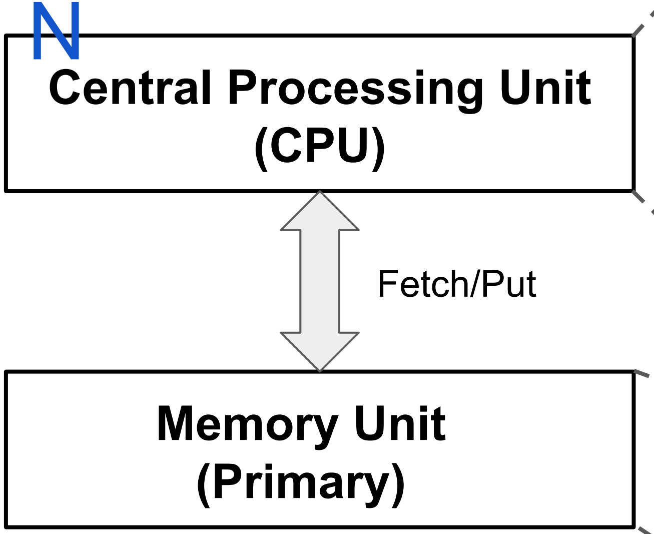
Each block represents the charge of a single capacitor in memory

0	0	1	0	0	0	1	0	“34”
0	0	0	1	0	1	1	1	“23”
0	0	1	0	0	0	0	0	“add(1,3)”
2^7	2^6	2^5	2^4	2^3	2^2	2^1		

2^0

Computer Architecture

COMPUTATIO N



Central Processing Unit (CPU)

Fetch/Put

Memory Unit (Primary)

1. Fetch instruction from memory location 2
2. Execute instruction:
   a. Load the first number into the CPU
   b. Add the second number to this number
3. Store the sum in memory location 4

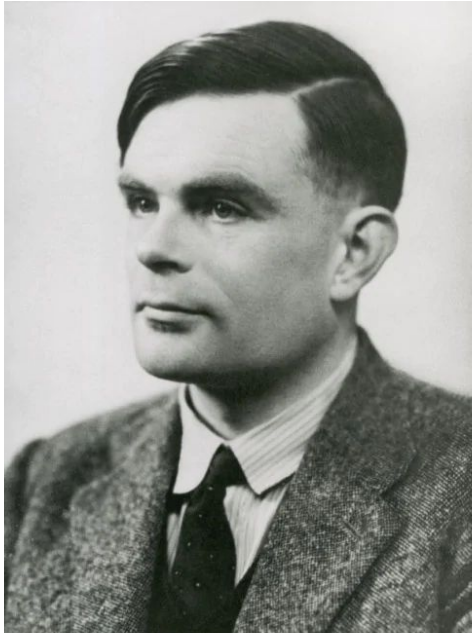
$$\begin{array}{rl} & 0\;0\;1\;0\;0^{1}\;0^{1}1\;0 \\ + & 0\;0\;0\;1\;0\;\;1\;1\;1 \\ \hline & 0\;0\;1\;1\;1\;\;0\;0\;1 \end{array}$$

1	2	3	4
Value: “23”	**Instruction:** “add(1,3)”	**Value:** “34”	**Value:** **“57”**
...	...	...	

Computer Architecture

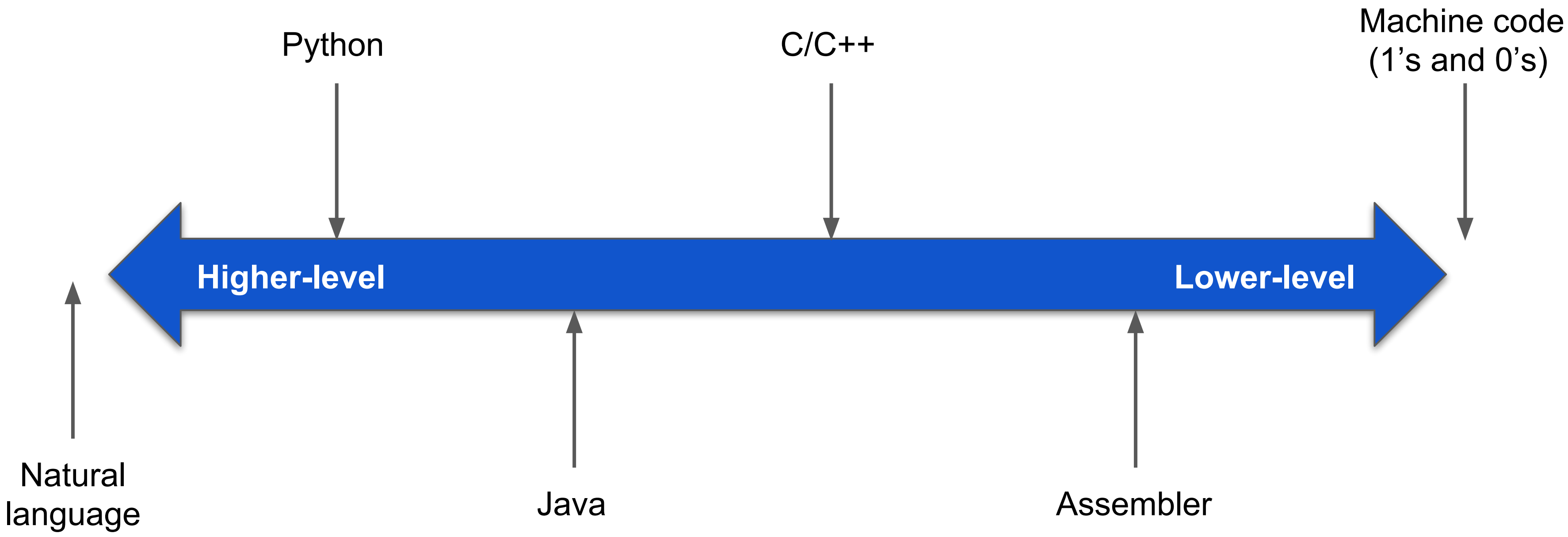
Modern computers using Turing’s model of computation

- Described by a Turing machine
- Any algorithm that you can conceive of and execute as a human being, can be represented as a set of operations in a Turing machine.

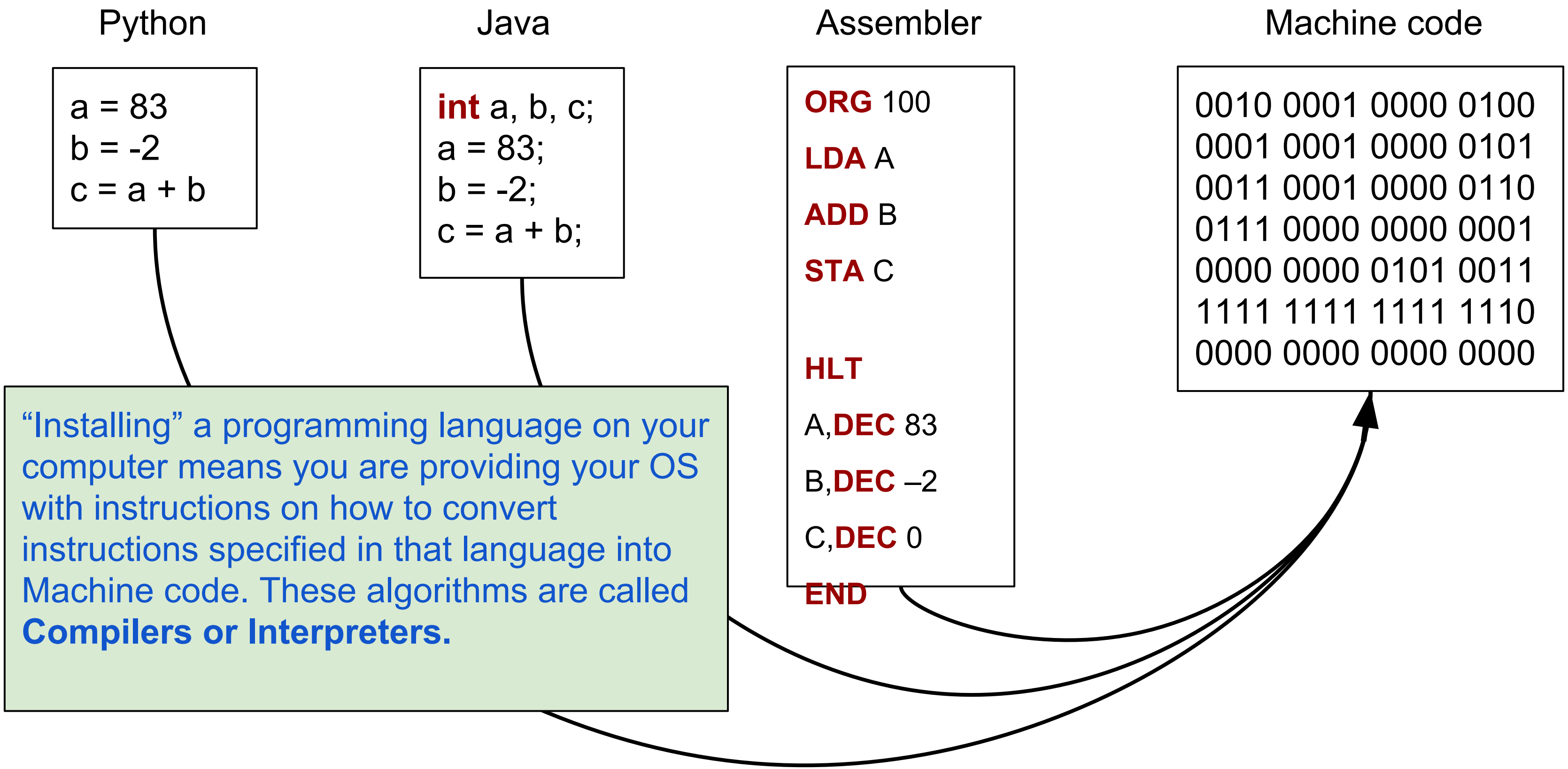


Example (square root)

Programming languages



Programming languages



Python

```
a = 83
b = -2
c = a + b
```

Java

```
int a, b, c;
a = 83;
b = -2;
c = a + b;
```

Assembler

```
ORG 100
LDA A
ADD B
STA C

HLT
A,DEC 83
B,DEC –2
C,DEC 0
END
```

Machine code

```
0010 0001 0000 0100
0001 0001 0000 0101
0011 0001 0000 0110
0111 0000 0000 0001
0000 0000 0101 0011
1111 1111 1111 1110
0000 0000 0000 0000
```

“Installing” a programming language on your computer means you are providing your OS with instructions on how to convert instructions specified in that language into Machine code. These algorithms are called **Compilers or Interpreters.**

Part 3

Examples of simple algorithms

Find the largest number in a list

Judge name	Number of cases
Ilešič	266
Jann	280
Schockweiler	251
Moitinho de Almeida	272
Toader	263
Rosas	284
Silva de Lapuerta	252
Gulmann	255

Find the largest number in a list

Number of cases
266
280
251
272
263
284
252
255

- Input: ?
- Output: ?
- Instructions: ?

Find the largest number in a list

Number of cases
266
280
251
272
263
284
252
255

- Input: list of numbers e.g. 266,280,251,272,263,284,252,255
- Output: largest number in the list e.g. 284
- Outline of algorithm:
    - Assume the first number in the list is the largest
    - Consider each number in the list one-by-one from top to bottom
    - While we do this, if the current number being looked at is larger than the current largest number, consider this number to be the new largest
    - Stop when we get to the end of the list

Sort a list of numbers from smallest to largest

Number of cases
266
280
251
272
263
284
252
255

- Input: ?
- Output: ?
- Instructions: ?

Sort a list of numbers from smallest to largest

Number of cases
266
280
251
272
263
284
252
255

- Input: list of numbers e.g. 266,280,251,272,263,284,252,255
- Output: list of numbers sorted ascending order e.g. 251,252,255,263,266,272,280,284
- Instructions: ?

Part 4

Examples of ubiquitously reused algorithms

Part 5

Ethical significance of algorithms

“Algorithms are not arbiters of objective truth and fairness simply because they're math.” - **Zoe Quinn**

Summary

- Algorithms are step-by-step instructions for executing a task
- Issuing algorithmic instructions to computers requires a special language:
  - Computers, on their lowest-level, are only able to manipulate data represented in a binary language (0’s and 1’s)
  - Humans have created ‘high-level’ programming languages to help us communicate with computers more concisely
  - Compilers (algorithms for translating instructions from ‘high-level’ programming languages into computer language) make ‘high-level’ communication with computers possible
- Algorithms are usually developed in a modular way so they can be reused in other algorithms

Summary (cont.)

- Algorithms can be classified according to the problems they try to solve as well as the procedures they use to solve them
- Ethical issues sometimes arise with developing algorithms:
    - Issues can stem from lack of transparency in the algorithm’s steps (e.g. Machine Learning)
    - They can also stem from fundamental questions of whether algorithms (and automation in general) should be applied to solve certain problems (social implications etc.)