How stuff works in Computer Science

Giselle Reis

15-110 Principles of Computing

Spring 2020

Disclaimer

The goal of this lecture is to give an overview of how some of the fundamental aspects of computer science. The topics chosen are the ones that underlying ubiquitous computational tools used on a daily basis.

The topics we will cover are:

- Computers
- Programs
- Internet
- Artificial intelligence

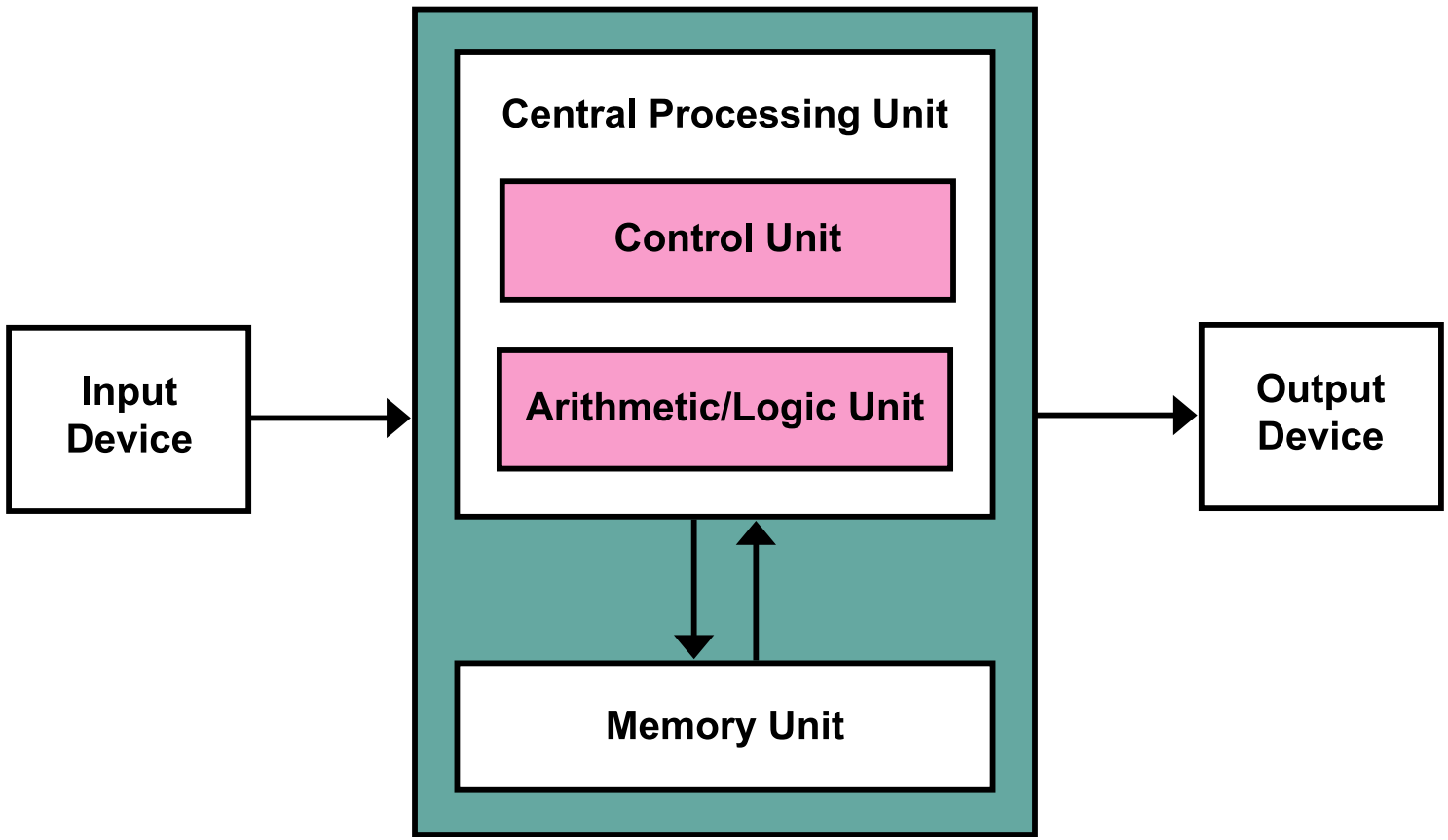
Computers

We use computers on a daily basis, whether this is our laptops, desktops, phones, cars, or watches. But how do things work under the hood?

Computers

Architecture

Most computers follow the Von Neumann architecture (1945):



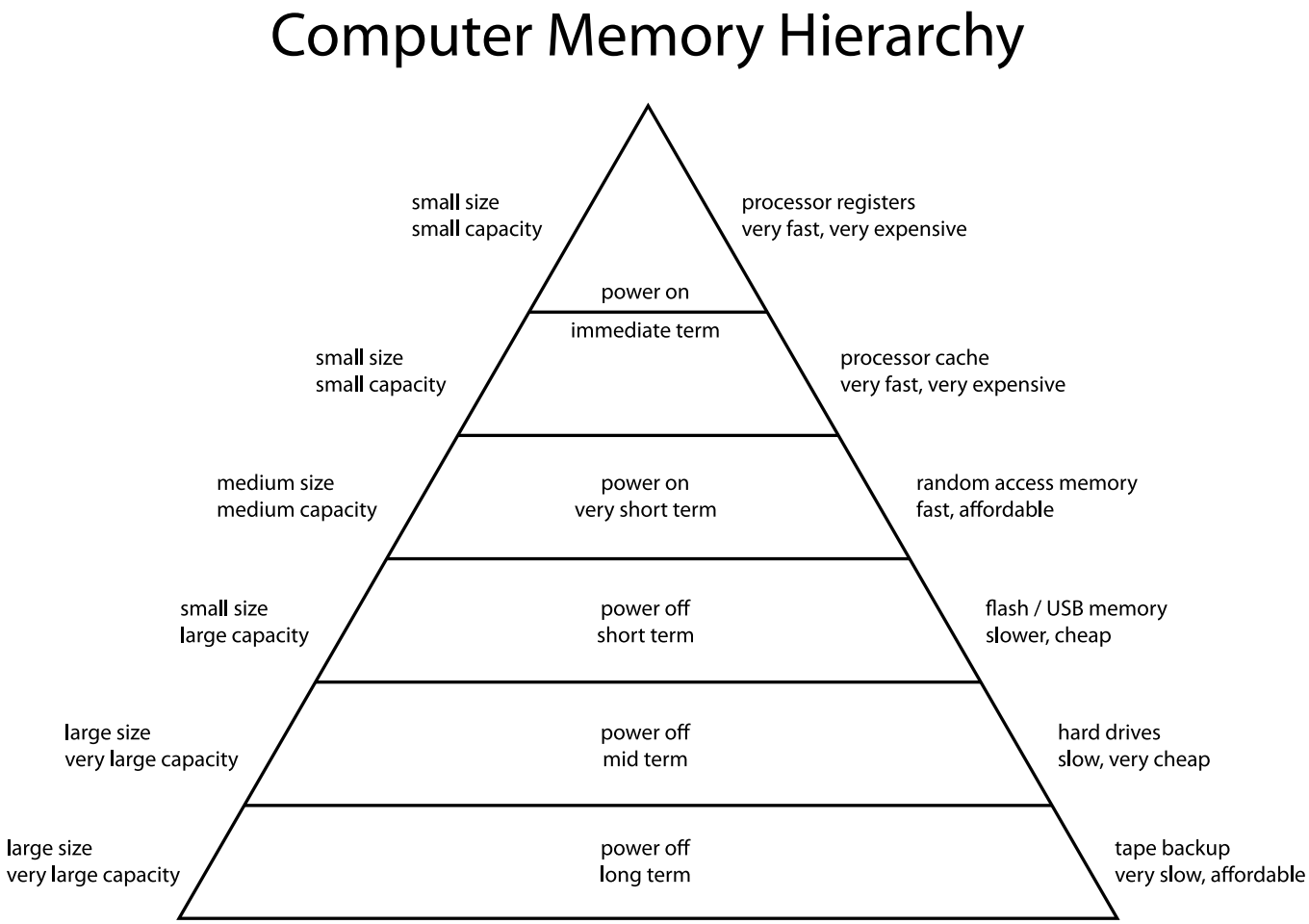
Computers

Components

- Input devices (e.g. keyboard, mouse)
- Output devices (e.g. monitor, sound card, printer)
- Memory unit:
  - RAM: random access memory – volatile
  - Hard disk: persistent
  - ROM: read only memory – persistent (e.g. video games)
  - Memory hierarchy depends on access speed (cache)
- CPU: central processing unit
  - Control unit: instruction register and program counter
  - Arithmetic/Logic unit: where the magic happens

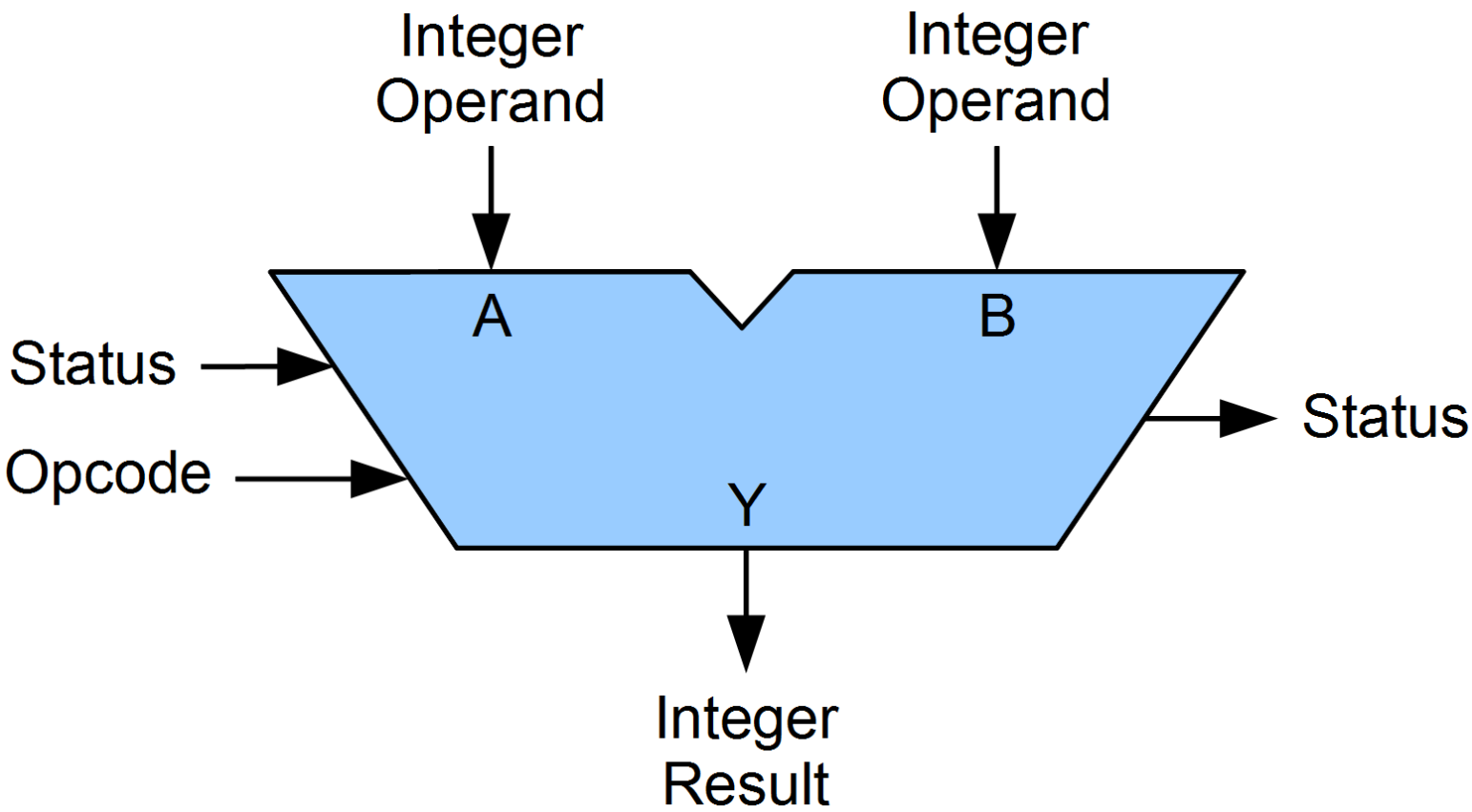
Computers

Types of memory



Computers

ALU



Programs

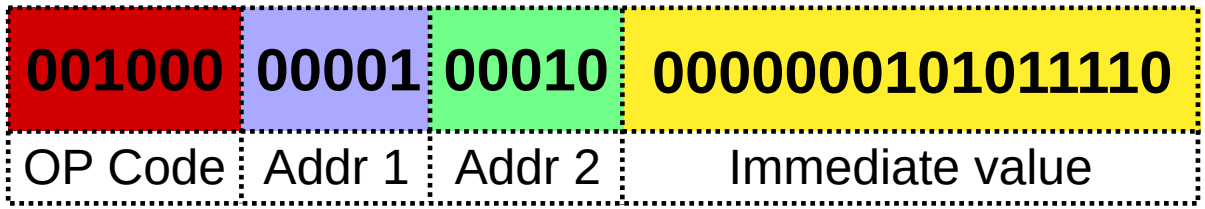
You know how to program, but how does the computer translate your instructions into a program that does stuff, *really*?

Programs

Machine Language

The computer only understands one language: **machine language** (also called *assembly*).

MIPS32 Add Immediate Instruction



Equivalent mnemonic: **addi $r1, $r2, 350**

Each instruction is executed by the ALU.

Language specification:
`https://user.eng.umd.edu/~blj/RiSC/RiSC-isa.pdf`

Programs

Compilers

- Compilers are programs that translate source code into machine code.
- The result of this translation is the *executable* file.
- C++ example:

  ```
  obdump -d a.out
  hexdump -C a.out
  ```

- Machine dependency.

Programs

Compilers

Compilation happens in several steps.

1. Syntax analysis
   - Lexing: identifies tokens
   - Parsing: identifies expressions
   - Result: abstract syntax tree
   - `SyntaxError: invalid syntax`
2. Semantic analysis
   - Type checking
   - Symbols (are all of them defined?)
   - `NameError: name 'a' is not defined`
3. Code generation

We don't actually get to see Python's machine code because we use it as an *interpreted* language (sort of an interactive on-the-fly compilation + execution).

Internet

We cannot imagine our lives without internet nowadays. What is the infrastructure that allows us to communicate to the other side of the world?

Internet

Protocols

If we connect two computers, they can send signals to each other through a cable. How do we guarantee that they speak the same language?

Via protocols!

Internet protocols are organized by layers:

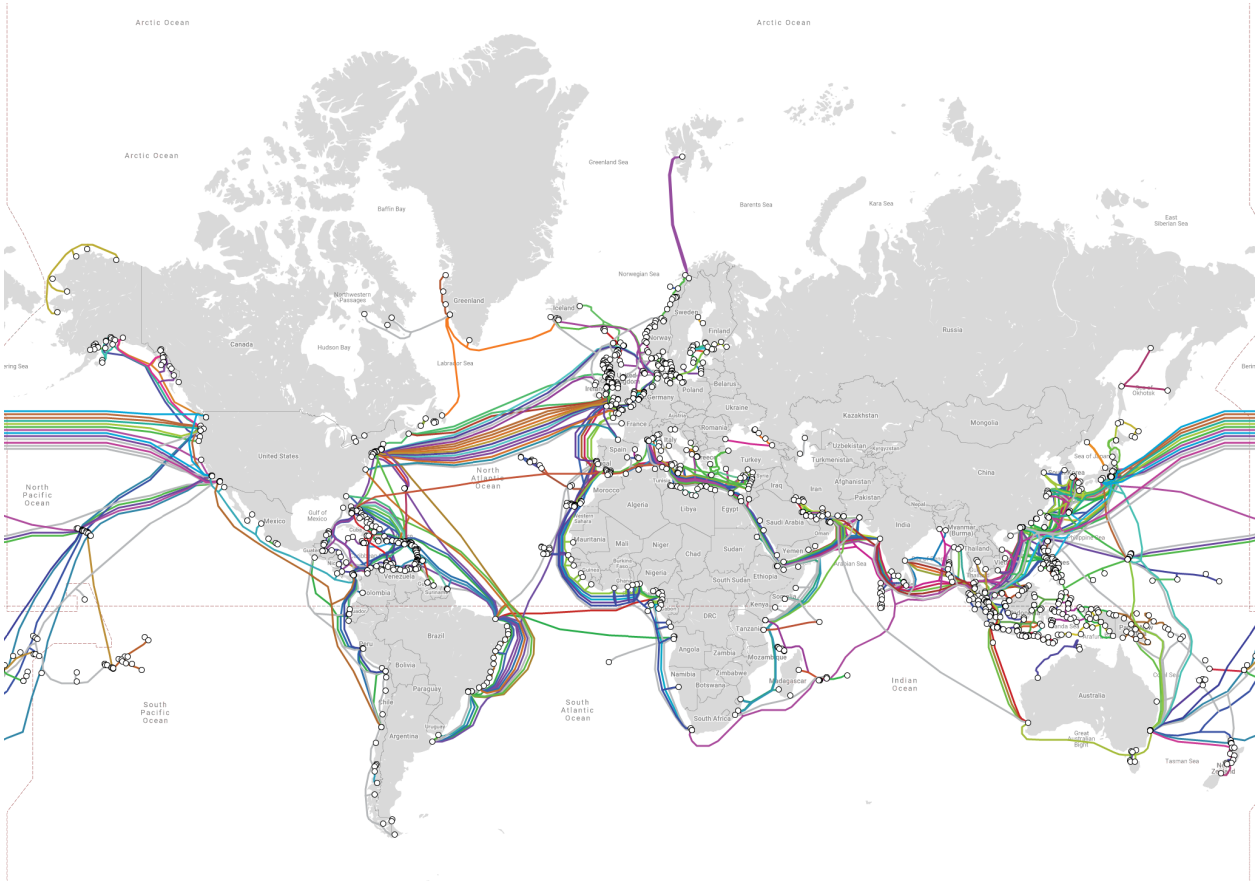
- Application (e.g. HTTP – websites, IMAP – e-mail, SSH – how you connect to CMU's servers, FTP – for file transfer)
- Transport (e.g. TCP)
- Internet (e.g. IP – defines who you are in the network)
- Link (e.g. ethernet, WiFi)

`wireshark`

Internet

Connections

How are all computers in the world connected? Submarine cables![1]

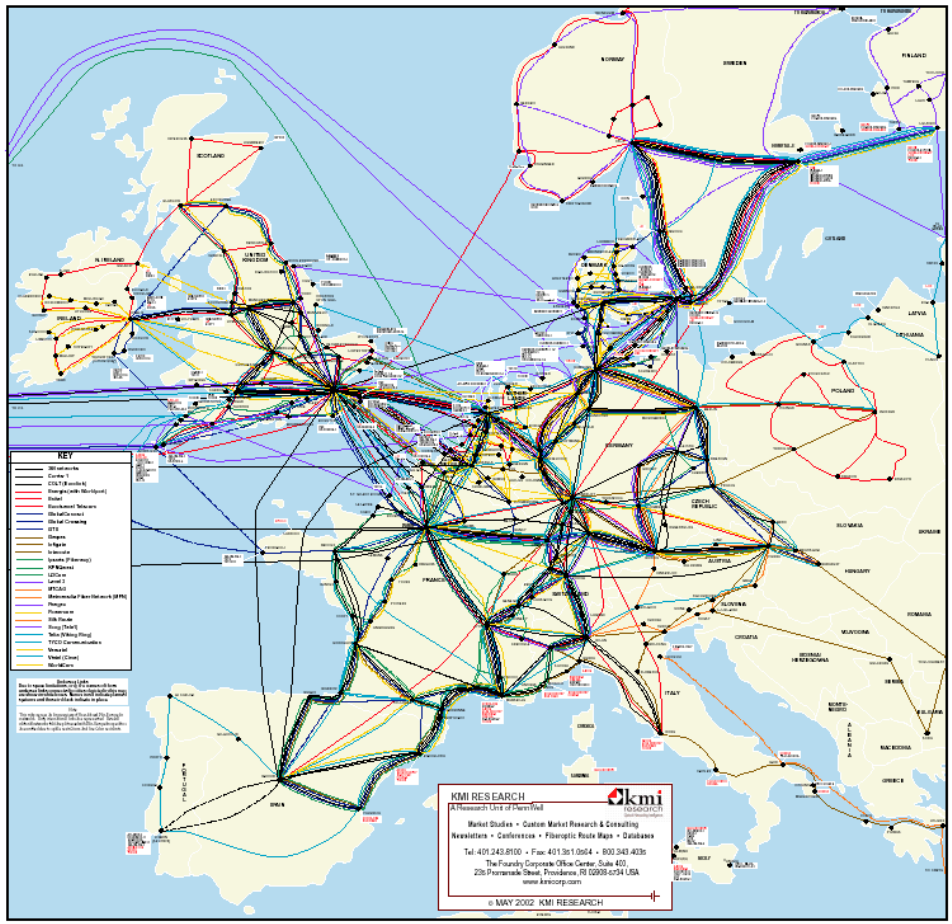


[1] https://www.submarinecablemap.com/

Internet

Connections

And fiber optic cables in countries and regions.

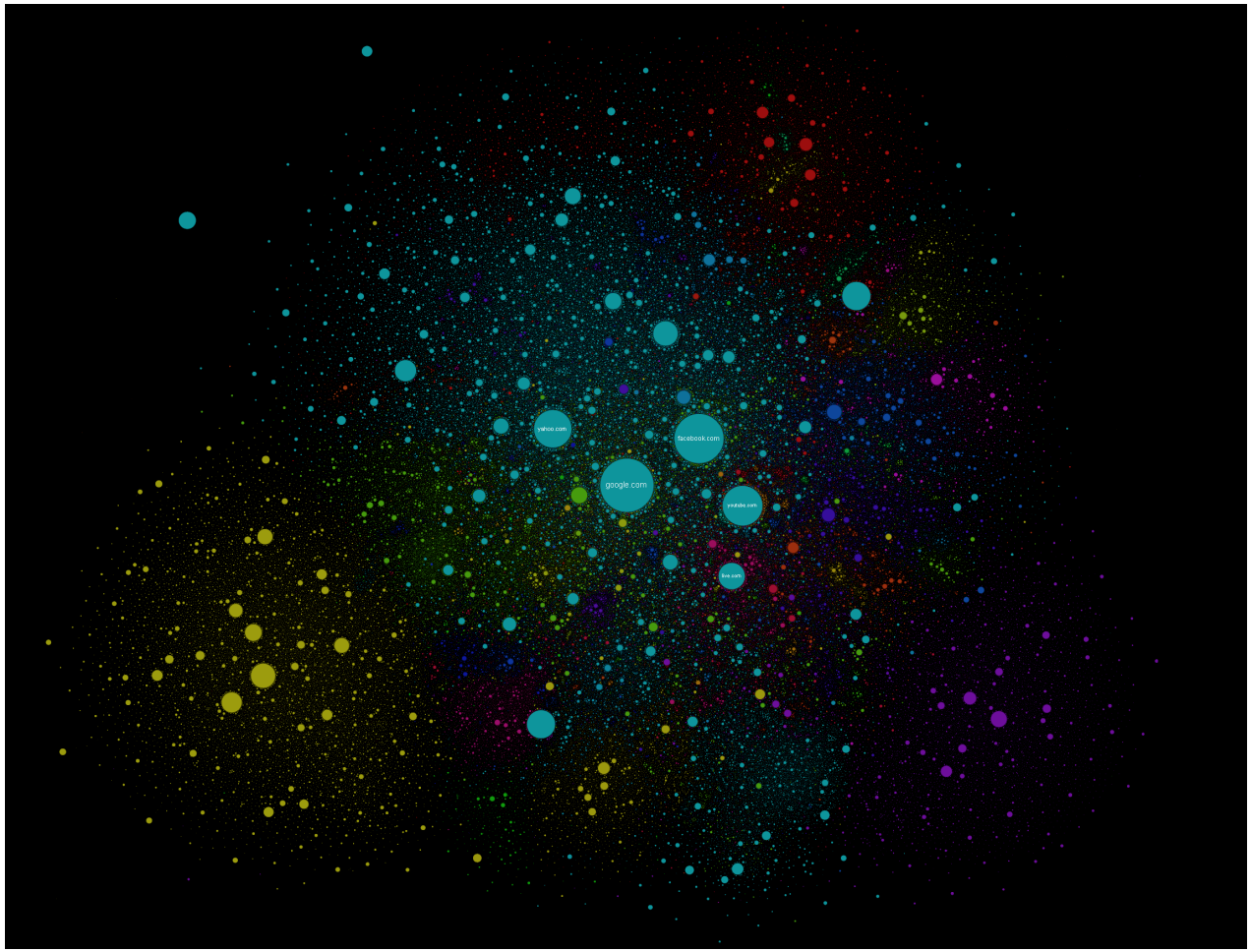


Routers direct the traffic where it needs to go.

Internet

Websites

In our eyes, the internet can be seen as a collection of websites.



`https://internet-map.net/`

Artificial intelligence

Now that you know how a computer works, do you think they can be *intelligent*?

Artificial intelligence

Overview

AI is a traditional field in computer science (since, at least, the 40s).

It comprises of many (enourmous) subfields:

- Machine learning
- Natural language processing
- Automated reasoning
- ... etc.

N.B.: These subfields are not disjoint.

Artificial intelligence

Turing test

- Proposed in 1950
- Can a computer be undistinguishable from a human, for another human?
- A lot of technical attempts (e.g. Eliza)
- A lot of philosophical debate (e.g. how does human and artificial "intelligence" relate?)

`https://www.cleverbot.com/`

Artificial intelligence

More pragmatic

Machine learning:

- Predicting outcomes from known data
- More data $\rightarrow$ better predictions? (not always)
- Methods are heavily based on maths and statistics

Example: supervised learning (regression)

Artificial intelligence

More pragmatic

Natural language processing:

- How to make computers process and understand our languages
- Intersection between computer science and linguistics
- Approaches: symbolic or statistical

Examples: translation, speech recognition, text generation

Time flies like an arrow; fruit flies like a banana.

Artificial intelligence

More pragmatic

Automated reasoning[2]:

- Get computers to prove stuff
- We need to translate human reasoning into something mechanical (mechanizing logic)
- Extremely hard problem in its most generality
- Specialized techniques can work well

Examples: four colour theorem, Kepler conjecture (sphere packing)

[2] What I work with :)

Thank you, good luck for finals, and enjoy the summer ☺