



MASTER IN ENTREPRENEURSHIP  
INNOVATION MANAGEMENT  
IN COLLABORATION WITH **MIT SLOAN**

IN COLLABORATION WITH

**MIT** MANAGEMENT  
SLOAN SCHOOL



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**PARTHENOPE**

MASTER MEIM 2022-2023

# Some considerations on the current state of Digital Technologies /Strategies and Artificial Intelligence.

## What you will learn in the AI, DS and DT courses

### Lesson 1

Prof. Giulio Giunta

Prof. of Scientific Computing, Università degli Studi di Napoli Parthenope

## a day to remember: November 30, 2022

San Francisco – based OpenAI launches ChatGPT

as of December 4, 2022, ChatGPT already had over one million users

in January 2023, ChatGPT reached over 100 million users, making it the fastest growing consumer application to date

the beginning of a new era for Information Technology

Information Technology or **Intelligent Transformation** ?

## ChatGPT on Wikipedia:

ChatGPT is an artificial intelligence chatbot

it is built on top of OpenAI's GPT-3 family of **large language models** and has been fine-tuned (an approach to transfer learning) using both supervised and reinforcement learning techniques

GPT stands for **pre-trained generative transformer**

<https://chat.openai.com>



## ChatGPT on Wikipedia:

although the core function of a chatbot is to mimic a human conversationalist, ChatGPT is versatile

it can write and debug computer programs, compose music, teleplays, fairy tales, and student essays; answer test questions; write poetry and song lyrics; emulate a Linux system; simulate an entire chat room; play games like tic-tac-toe; and simulate an ATM

ChatGPT's training data includes man pages and information about internet phenomena and programming languages, such as bulletin board systems and the Python programming language

<https://chat.openai.com>



## ChatGPT prompt

**tell me about chatGPT training data sources**

## ChatGPT's answer

1. Web pages: OpenAI used web crawling techniques to collect a large amount of text data from publicly available web pages. This includes news articles, blog posts, and other types of online content
2. Books: OpenAI also used a large corpus of books, including both fiction and non-fiction works, as a source of training data
3. Academic papers: The training data for ChatGPT includes a large collection of academic papers from various fields, including computer science, linguistics, and other scientific disciplines
4. Wikipedia: OpenAI used Wikipedia articles as a source of training data. Wikipedia provides a rich source of information on a wide range of topics and is a valuable resource for natural language processing tasks
5. Conversational data: In addition to written text, OpenAI also used conversational data as a source of training data. This includes chat logs, email exchanges, and other types of online conversations.

## ChatGPT prompt

**I will teach a 3-hour introductory lecture to a Master's degree in Entrepreneurship and Innovation Management.**

**My lesson introduces 3 courses on digital technologies, digital strategies and artificial intelligence.**

**I also want to draw the current state of IT.**

**How would you structure my introductory lesson?**

## ChatGPT's answer

Great! Here's a possible structure for your introduction lecture:

### I. Introduction

Provide an overview of the lecture's objectives and structure

### II. The current state of IT

Define IT and its history

Discuss the latest trends and developments in IT, such as cloud computing, big data analytics, and the Internet of Things

Explain how these technologies are transforming various industries and changing the way businesses operate

Highlight the potential benefits and challenges of using IT in entrepreneurship and innovation management

## ChatGPT's answer

### III. Digital technologies in entrepreneurship and innovation management

Discuss the importance of digital technologies, such as social media, mobile devices, and e-commerce, in entrepreneurship and innovation management

Explain how digital technologies can be leveraged for competitive advantage, customer engagement, and product innovation

Highlight some of the challenges and risks associated with digital technologies, such as cybersecurity threats and ethical concerns

Provide examples of successful digital strategies in entrepreneurship and innovation management



## ChatGPT's answer

### IV. Digital strategies in entrepreneurship and innovation management

Define digital strategy and its importance in today's business environment

Discuss the key elements of a successful digital strategy, such as customer focus, data-driven decision-making, and agile innovation

Explain how digital strategy can help entrepreneurs and innovators identify new opportunities, create value, and outmaneuver competitors

Provide examples of successful digital strategies in entrepreneurship and innovation management

## ChatGPT's answer

### V. Artificial intelligence in entrepreneurship and innovation management

Define artificial intelligence and its various forms, such as machine learning, natural language processing, and robotics

Discuss the potential applications of artificial intelligence in entrepreneurship and innovation management, such as predictive analytics, chatbots, and autonomous systems

Provide examples of how artificial intelligence can be used to solve complex problems and make more informed decisions

Highlight some of the challenges and risks associated with artificial intelligence, such as bias and job displacement

## ChatGPT's answer

### VI. Conclusion

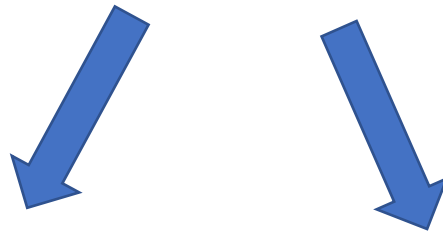
Summarize the key points covered in the lecture

Reiterate the importance of staying up-to-date with the latest developments in IT and how they can be leveraged for entrepreneurial and innovative purposes

Invite questions and comments from the audience

## Organization of the Artificial Intelligence course (72 hours)

### ARTIFICIAL INTELLIGENCE



#### ARTIFICIAL INTELLIGENCE

History and taxonomies

Basic tools for Computational intelligence

Computer vision

#### MACHINE LEARNING

Neural Networks

Supervised and unsupervised training

Clustering, feature selection, dimensionality reduction

## Organization of the Digital Tech course (116 hours)

### DIGITAL TECH



#### CLOUD COMPUTING

(with a short course on coding in Python)

#### HIGH PERFORMANCE COMPUTING

(with a short course on computational modeling in finance)

Amazon Web Services Academy “Cloud Foundation” course (enabling the students to apply for the first AWS certificate)

## Organization of the Digital Strategy course (40 hours)

### DIGITAL STRATEGY



Recommendation Systems: user profiling, ranking

Internet of Things (IoT): how can we benefit from this technology ?

Security issues in Cloud Computing

Economic perspectives of AI and ML

EU policy and regulation on the use of AI and ML

Almawave SpA: Artificial Intelligence: the impacts on business, technologies and use cases

with a short course on Machine Learning programming on Apple devices such as iPhone iOS, watchOS and tvOS

## Try this

`meteo@uniparthenope.it`

the meteo/ocean forecast service at UniParthenope (also an app for your mobile)

## ChatGPT's prompt

Act as a meteorologist. Weather report about bay of Naples, Italy. Use data from `meteo@uniparthenope.it`. Use knots for wind and Celsius for temperature. Write the report in Italian.

look at **ChatGPT's answer**



## Where does ChatGPT come from?

### The long journey from ELIZA to ChatBPT

- the first natural language processing computer program ELIZA, created from 1964 to 1966 at MIT by Joseph Weizenbaum
- Expert systems (1970s-1980s)
- Statistical methods in NLP (1990s)
- Deep Neural Networks (late 2000s)
- Natural language processing chatbots and virtual assistants, such as Apple's Siri and Amazon's Alexa (2010s)



## Where does ChatGPT come from?

### The long journey from ELIZA to ChatBPT

- Google's development of the Transformer architecture revolutionized natural language processing with the introduction of the concept of attention (2017)
- OpenAI's GPT-3 (Generative Pre-trained Transformer 3) was introduced, which is a state-of-the-art natural language processing model capable of generating human-like text responses and is used in ChatGPT (2020s)

## Where does ChatGPT come from?

ELIZA, running the DOCTOR script, was created to provide a parody of "the responses of a non-directional psychotherapist in an initial psychiatric interview"

```
Welcome to
```

```
EEEEEE LL      IIII  ZZZZZZ  AAAAA  
EE      LL      II    ZZ     AA  AA  
EEEEEE LL      II    ZZZ    AAAAAAA  
EE      LL      II    ZZ     AA  AA  
EEEEEE LLLLLL IIII  ZZZZZZ  AA  AA
```

```
Eliza is a mock Rogerian psychotherapist.  
The original program was described by Joseph Weizenbaum in 1966.  
This implementation by Norbert Landsteiner 2005.
```

```
ELIZA: Is something troubling you ?  
YOU:   Men are all alike.  
ELIZA: What is the connection, do you suppose ?  
YOU:   They're always bugging us about something or other.  
ELIZA: Can you think of a specific example ?  
YOU:   Well, my boyfriend made me come here.  
ELIZA: Is it important to you that your boyfriend made you come here ?  
YOU:   He says I'm depressed much of the time.  
ELIZA: I am sorry to hear that you are depressed.  
YOU:   It's true. I am unhappy.  
ELIZA: Can you explain what made you unhappy ?  
YOU:
```

## Where does ChatGPT come from?

ELIZA, running the DOCTOR script, was created to provide a parody of "the responses of a non-directional psychotherapist in an initial psychiatric interview"

```
Welcome to
```

```
EEEEEE LL      IIII  ZZZZZZ  AAAAA  
EE      LL      II    ZZ     AA  AA  
EEEEEE LL      II    ZZZ    AAAAAAA  
EE      LL      II    ZZ     AA  AA  
EEEEEE LLLLLL IIII  ZZZZZZ  AA  AA
```

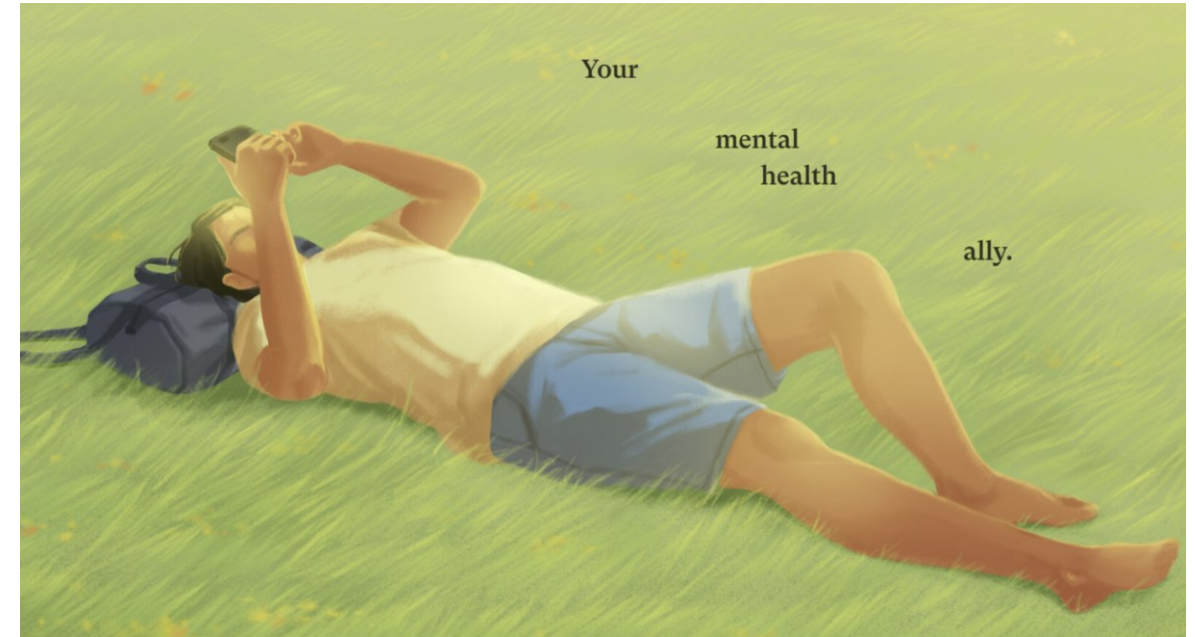
```
Eliza is a mock Rogerian psychotherapist.  
The original program was described by Joseph Weizenbaum in 1966.  
This implementation by Norbert Landsteiner 2005.
```

```
ELIZA: Is something troubling you ?  
YOU:   Men are all alike.  
ELIZA: What is the connection, do you suppose ?  
YOU:   They're always bugging us about something or other.  
ELIZA: Can you think of a specific example ?  
YOU:   Well, my boyfriend made me come here.  
ELIZA: Is it important to you that your boyfriend made you come here ?  
YOU:   He says I'm depressed much of the time.  
ELIZA: I am sorry to hear that you are depressed.  
YOU:   It's true. I am unhappy.  
ELIZA: Can you explain what made you unhappy ?  
YOU:
```

## A quite interesting App

ELIZA → WOEBOT

meet Woebot, your personal mental health ally that helps you get back to feeling like yourself. Grounded in science and powered by natural language processing, Woebot is easy to talk to and fits right into your life



 Woebot Health™

[woebothealth.com](https://woebothealth.com)

## Some other current AI tools

**Perplexity** generative AI chatbot

`perplexity.ai`

**Quilbot** paraphrase or re-write text

`quilbot.com`

**Speechify** text to speech

`speechify.com`

**DALL-E 2** text to image

`openai.com/product/dall-e-2.com`

**Github COPILOT** text to code

`github.com/features/copilot.com`

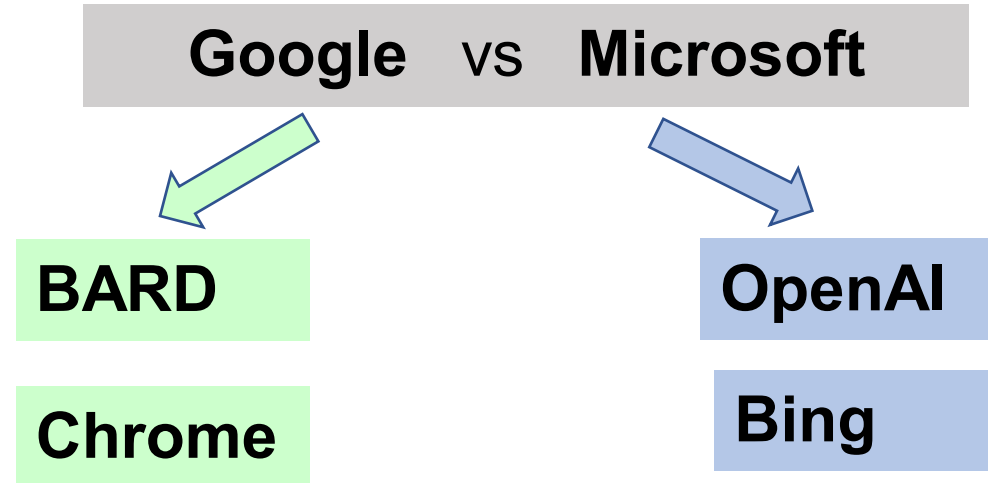
## Some other current AI tools

**LaMDA & Bard** generative AI chatbot by Google

**Sundar Pichai**, CEO of Google and Alphabet

<https://blog.google/technology/ai/bard-google-ai-search-updates//>

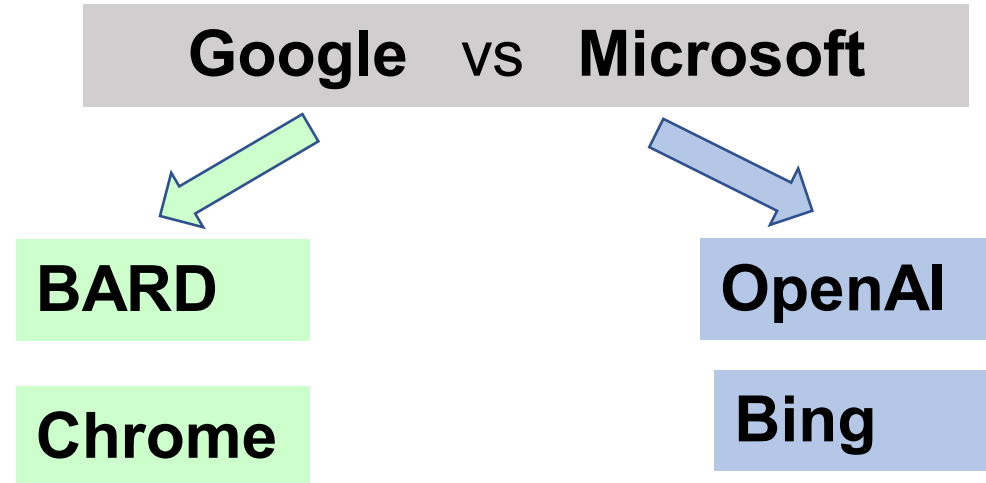
## An interesting comparison of business strategies



MIT Technology Review, March 2023

<https://www.technologyreview.com/2023/03/21/1070111/google-bard-chatgpt-openai-microsoft-bing-search/>

## An interesting comparison of business strategies



Harvard Business Review, February 2023

<https://hbr.org/2023/02/how-microsoft-became-innovative-again>



# GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou<sup>1</sup>, Sam Manning<sup>1,2</sup>, Pamela Mishkin\*<sup>1</sup>, and Daniel Rock<sup>3</sup>

<sup>1</sup>OpenAI

<sup>2</sup>OpenResearch

<sup>3</sup>University of Pennsylvania

March 21, 2023

[2303.10130.pdf \(arxiv.org\)](#)

[\[2303.10130\] GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models \(arxiv.org\)](#)

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

The first thing to explain is that what ChatGPT is always fundamentally trying to do is **to produce a reasonable continuation of whatever text it has got so far**, where by “reasonable” we mean “what one might expect someone to write after seeing what people have written on billions of webpages, books, etc.”

ChatGPT looks for words that in a certain sense “match in meaning” and the end result is that it produces a ranked list of words that might follow, together with **probabilities**

*The best thing about AI is its ability to*

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

the remarkable thing is that when ChatGPT does something like write an essay what it is essentially doing is just asking over and over again **“given the text so far, what should the next word be?”**—and each time adding a word

*The best thing about AI is its ability to*

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

But which one should it actually pick to add ?

One might think it should be the **highest-ranked** word (i.e. the one to which the highest **probability** was assigned)

*The best thing about AI is its ability to*

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

if we always pick the highest-ranked word, we will typically get a very “flat” essay, that never seems to show any **creativity** (and even sometimes repeats word for word). But if sometimes (**at random**) we pick lower-ranked words, we get a “more interesting” essay.

There is a particular so-called **temperature** parameter that determines how often lower-ranked words will be used, and for essay generation, it turns out that a **temperature** of 0.8 seems best

*The best thing about AI is its ability to*

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

where do those **probabilities** come from?

*The best thing about AI is its ability to*

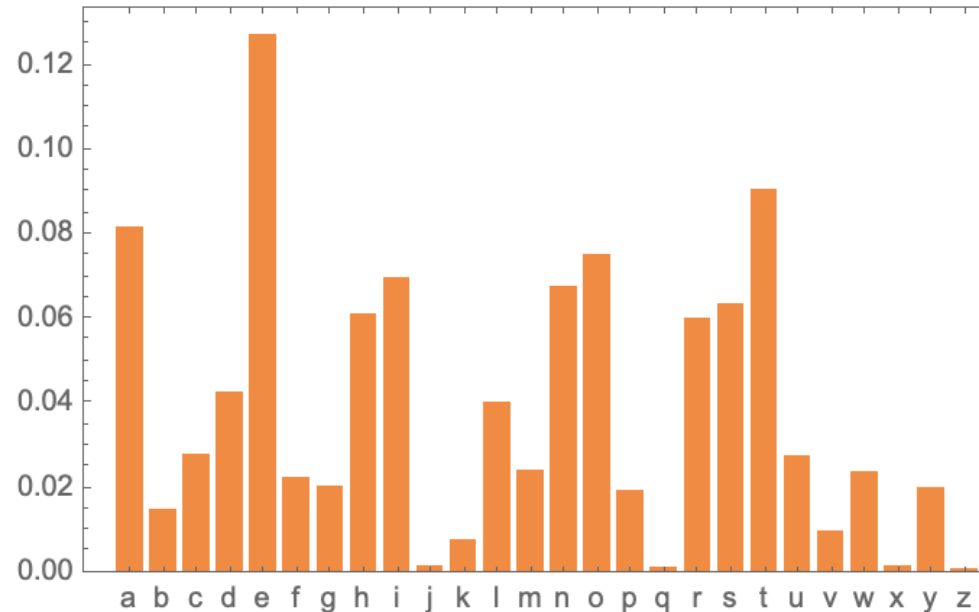
learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

where do those **probabilities** come from?

probability  
distribution  
of letters

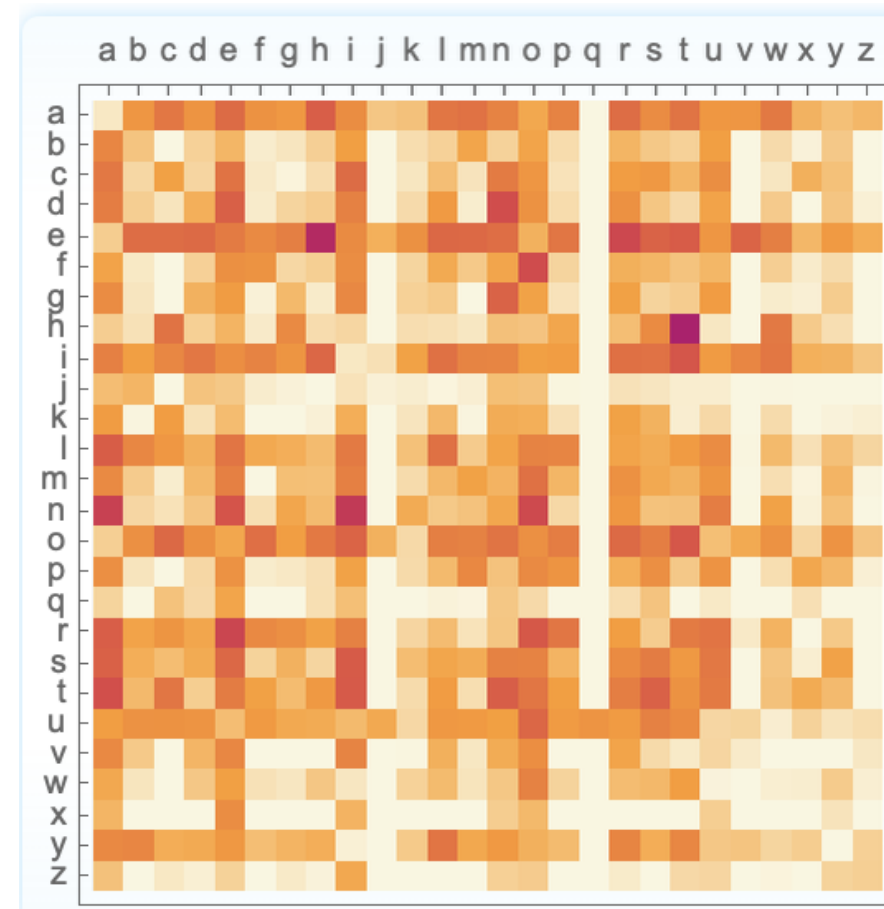


## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

where do those **probabilities** come from?

2-gram probability distribution of letters





## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

ChatGPT deals with whole words, not letters

There are about 40,000 reasonably commonly used words in English

And by looking at a large corpus of English texts (say a few million books, with altogether a few hundred billion words), we can get an estimate of how common each word is

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

And we might imagine that if we were able to use **sufficiently long n-grams** we would basically “get a ChatGPT” — in the sense that we would get something that would generate essay-length sequences of words with the “correct overall essay probabilities”

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

In a crawl of the web there might be a few hundred billion words; in books that have been digitized there might be another hundred billion words

But with 40,000 common words, even the number of possible 2-grams is already 1.6 billion, and the number of possible 3-grams is 60 trillion

So there is no way we can estimate the probabilities even for all of these from text that is out there

And by the time we get to “essay fragments” of 20 words, the number of possibilities is larger than the number of particles in the universe, so in a sense they could never all be written down

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

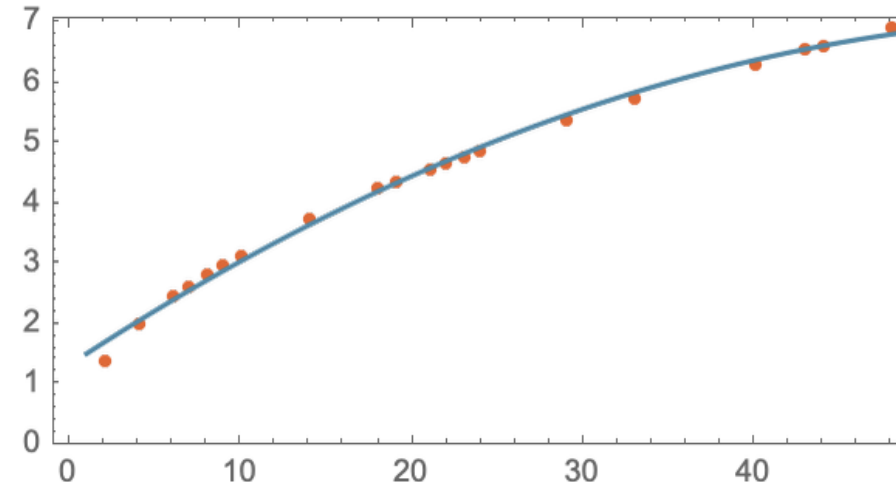
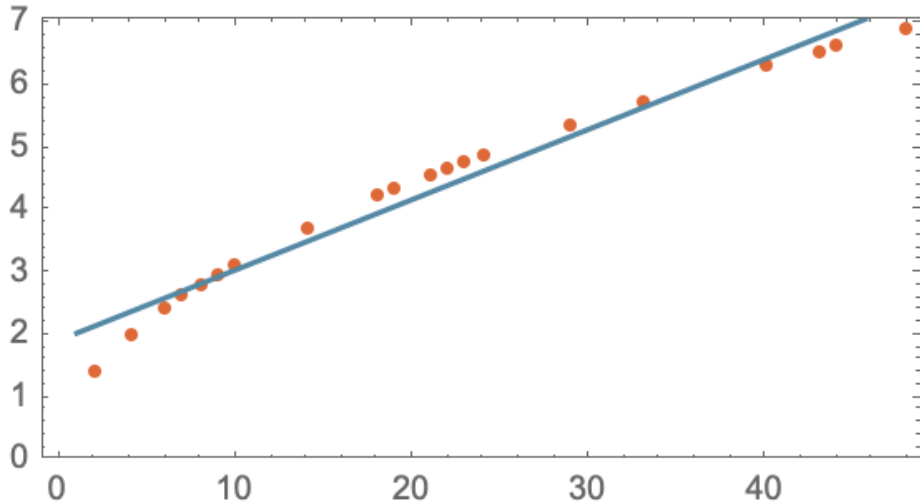
So what can we do?

The big idea is to make a **model** that lets us **estimate the probabilities** with which sequences should occur — even though we have never explicitly seen those sequences in the corpus of text we have looked at

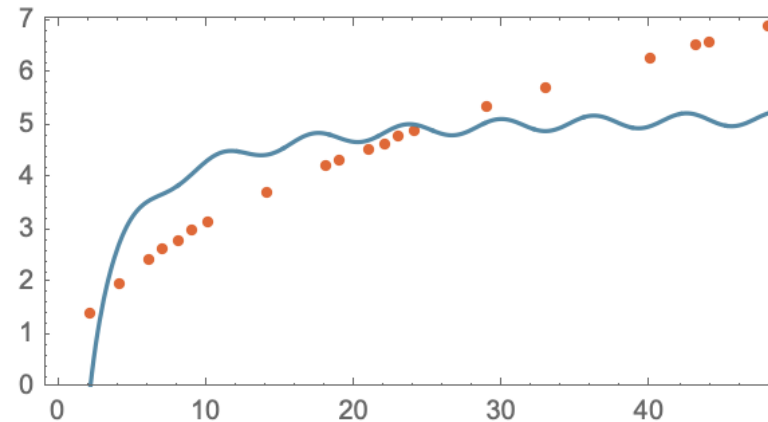
At the core of ChatGPT is precisely a so-called **Large Language Model** (LLM) that has been built to do a good job of estimating those probabilities

# Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)



simple models

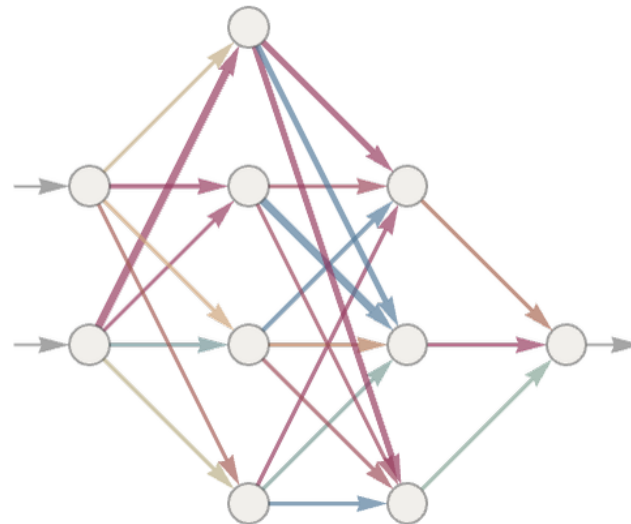


## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

Neural Networks are **general universal models**

(universal approximation theorem)

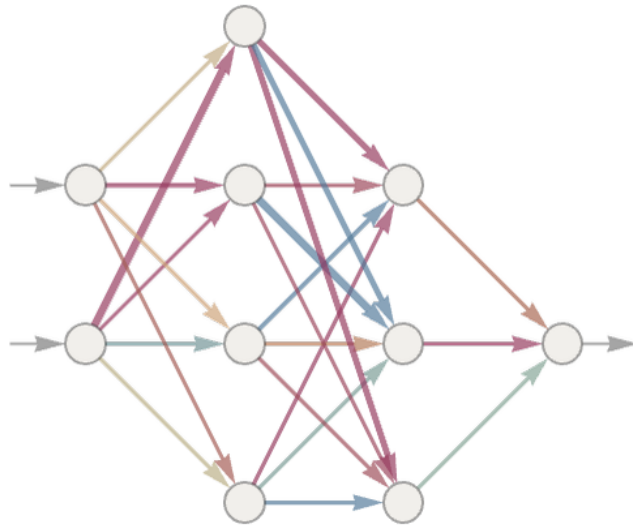


Neurons (nodes)  
and connections

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

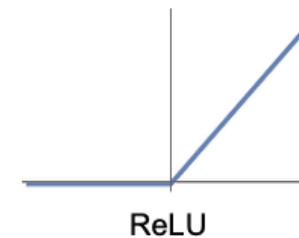
### Neural Network



Neurons (nodes)  
and connections

if a neuron has inputs  $x = \{x_1, x_2, \dots, x_n\}$  then it computes  $f(w \cdot x + b)$

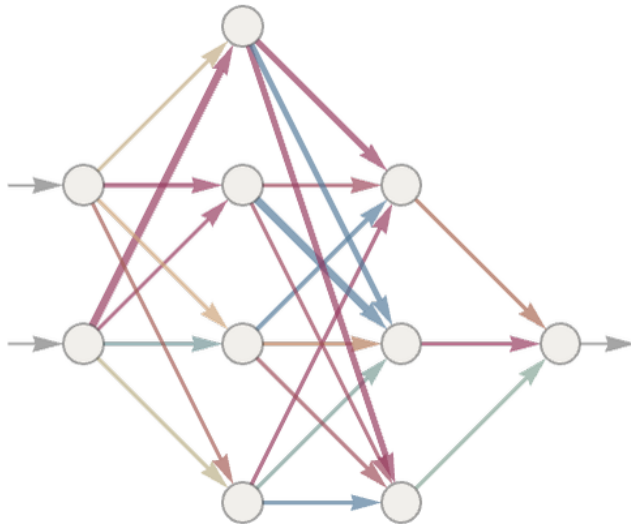
weights  $w$  and constant  $b$  are generally chosen differently for each neuron in the network; the function  $f$  is usually the same for all nodes



## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

### Neural Network



Neurons (nodes)  
and connections

what makes neural nets so useful (presumably also in brains) is that not only can they in principle do all sorts of tasks, but they can be incrementally “trained from examples” to do those tasks

technically, the training is a very large scale mathematical / computational problem:  
the **minimization** of a (loss) function of billions of variables



## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

Neural nets are fundamentally based on numbers

So if we are going to use them to work on something like text, we will need a way to represent our text with numbers

We could start (essentially as ChatGPT does) by just assigning a number to every word in the dictionary  
But there is an important idea — that is central to ChatGPT — that goes beyond that and it is the idea of “**embeddings**”

One can think of an embedding as a way to try to represent the “essence” of something by an array of numbers — with the property that “nearby things” are represented by nearby numbers

We can think of a word embedding as trying to lay out words in a kind of “**meaning space**” in which words that are somehow “nearby in meaning” appear nearby in the embedding

The actual embeddings that are used in ChatGPT involve large lists of numbers

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

Ultimately ChatGPT (**ChatGPT 3.5**) is a giant neural net — currently a version of the so-called GPT-3 network with 175 billion weights

In many ways this is a neural net very much like the other ones we have discussed. But it is a neural net that is particularly set up for dealing with language

Its most notable feature is a piece of neural net architecture called a “**transformer**”

400 (core) layers involved, in some ways not a huge number  
But there are millions of neurons, with a total of 175 billion connections  
and therefore 175 billion weights

## Basic ideas behind ChatGPT

(from S. Wolfram “What Is ChatGPT Doing ... and Why Does It Work?”, February 14, 2023)

a transformer operates in three basic stages:

**First**, it takes the sequence of tokens that corresponds to the text so far, and finds an embedding (i.e. an array of numbers) that represents these

**Then** it operates on this embedding — in a “standard neural net way”, with values “rippling through” successive layers in a network — to produce a new embedding (i.e. a new array of numbers)

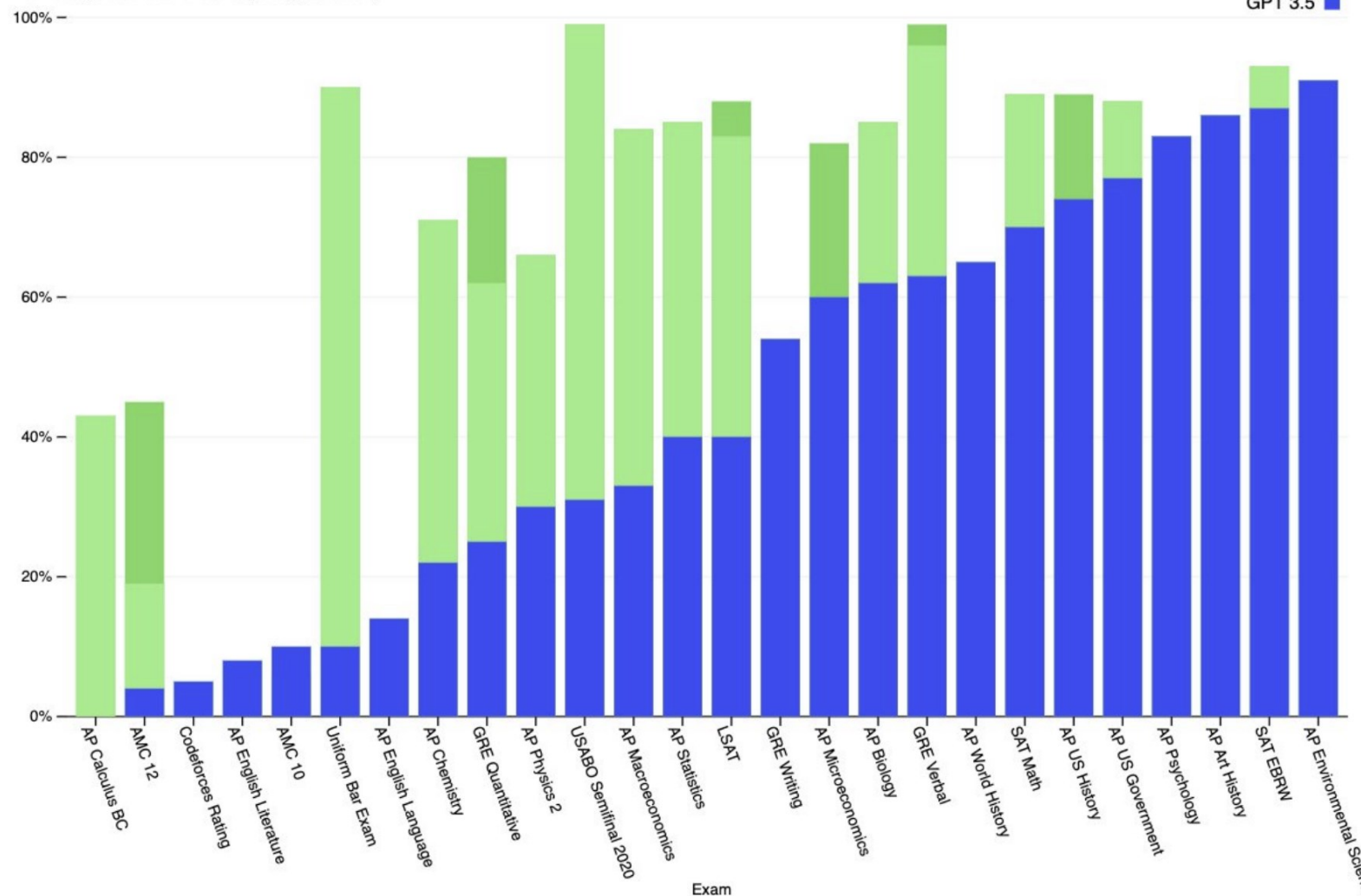
**Finally** it takes the last part of this array and generates from it an array of about 50,000 values that turn into probabilities for different possible next tokens

(and, yes, it so happens that there are about the same number of tokens used as there are common words in English, though only about 3000 of the tokens are whole words, and the rest are fragments)

**Exam results (ordered by GPT 3.5 performance)**

Estimated percentile lower bound (among test takers)

GPT 4 ■  
GPT 4 (no vision) ■  
GPT 3.5 ■



**SAT** Scholastic Assessment Test

**ACT** American College Testing

**AMC** American Mathematics Competitions

(8,10,12) american secondary school

**AP** Advanced Placement

**USABO** USA Biolympiad Test - Biology

**LSAT** Law School Admission Test

**UBE** Uniform Bar Examination, knowledge and skills

for lawyers

**GRE** General Test Comparison Tool for Law

Schools

**Codeforce Rating** platform for competitive

programmers

on March 14, 2023, OpenAI released **GPT-4**, a large **multimodal** model  
(accepting image and text inputs, emitting text outputs)

<https://openai.com/research/gpt-4>

## Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck    Varun Chandrasekaran    Ronen Eldan    Johannes Gehrke  
Eric Horvitz    Ece Kamar    Peter Lee    Yin Tat Lee    Yuanzhi Li    Scott Lundberg  
Harsha Nori    Hamid Palangi    Marco Tulio Ribeiro    Yi Zhang

Microsoft Research

<https://arxiv.org/pdf/2303.12712.pdf>

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Our approach to studying GPT-4's intelligence . . . . .	7
1.2	Organization of our demonstration . . . . .	8
<b>2</b>	<b>Multimodal and interdisciplinary composition</b>	<b>13</b>
2.1	Integrative ability . . . . .	13
2.2	Vision . . . . .	16
2.2.1	Image generation beyond memorization . . . . .	16
2.2.2	Image generation following detailed instructions (à la Dall-E) . . . . .	17
2.2.3	Possible application in sketch generation . . . . .	18
2.3	Music . . . . .	19
<b>3</b>	<b>Coding</b>	<b>21</b>
3.1	From instructions to code . . . . .	21
3.1.1	Coding challenges . . . . .	21
3.1.2	Real world scenarios . . . . .	22
3.2	Understanding existing code . . . . .	26

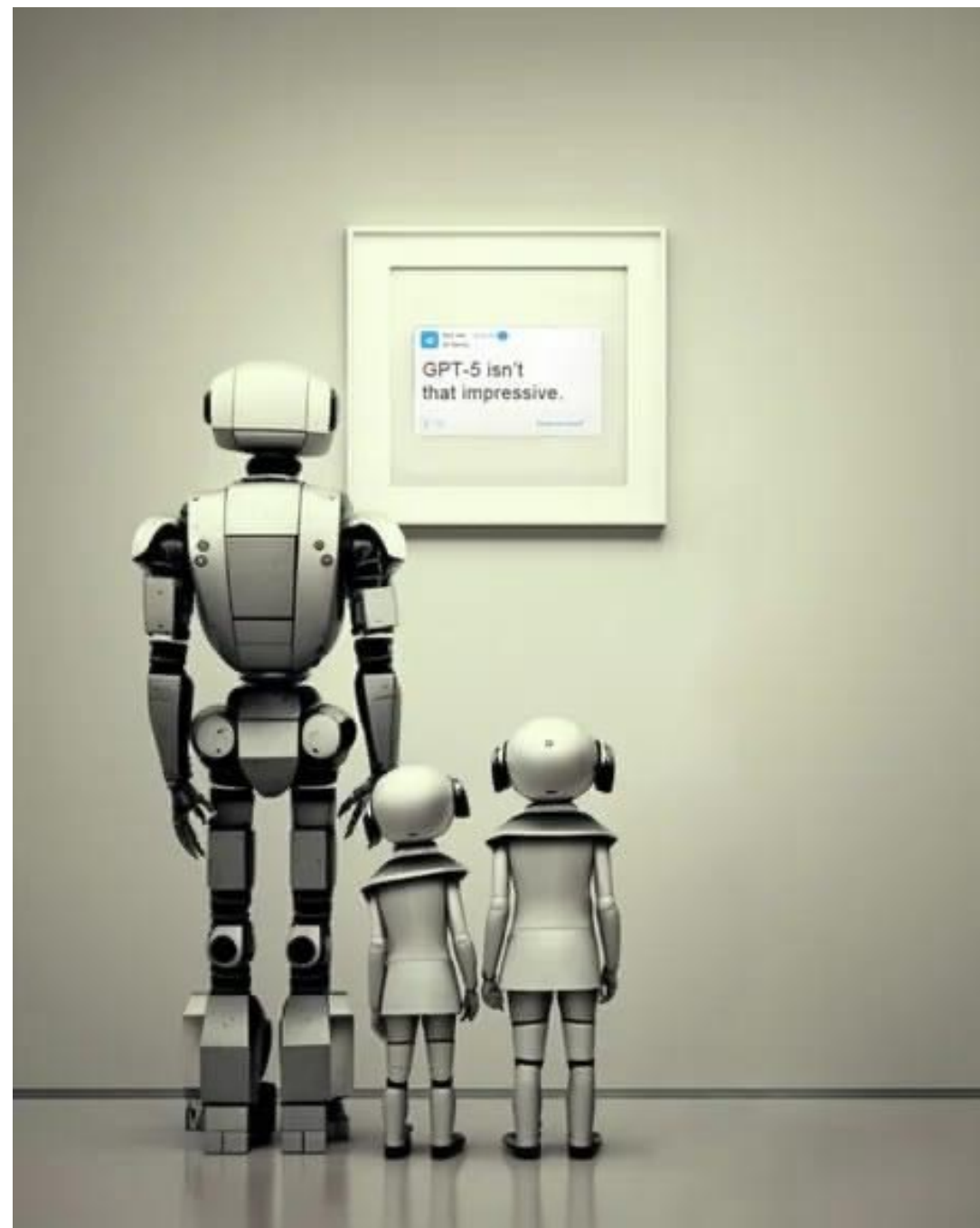
<b>4</b>	<b>Mathematical abilities</b>	<b>30</b>
4.1	A mathematical conversation with GPT-4 . . . . .	31
4.1.1	A first generalization of the original question . . . . .	31
4.1.2	A second variant of the original question . . . . .	32
4.1.3	Analysis of the limitations highlighted by conversation . . . . .	34
4.2	Performance on mathematical problem datasets . . . . .	35
4.3	Mathematical modeling in various domains . . . . .	37
4.4	Higher level mathematics . . . . .	39
<b>5</b>	<b>Interaction with the world</b>	<b>43</b>
5.1	Tool use . . . . .	43
5.1.1	Using multiple tools to solve more complex tasks . . . . .	44
5.1.2	Discussion . . . . .	49
5.2	Embodied Interaction . . . . .	49
5.2.1	Warmup: navigating a map . . . . .	49
5.2.2	Text-based games . . . . .	49
5.2.3	Real world problems . . . . .	52
5.2.4	Discussion . . . . .	53

<b>6</b>	<b>Interaction with humans</b>	<b>54</b>
6.1	Understanding Humans: Theory of Mind . . . . .	54
6.1.1	Testing specific aspects of theory of mind . . . . .	54
6.1.2	Testing theory of mind in realistic scenarios . . . . .	54
6.1.3	Discussion . . . . .	60
6.2	Talking to Humans: Explainability . . . . .	60
<b>7</b>	<b>Discriminative Capabilities</b>	<b>69</b>
7.1	PII Detection . . . . .	69
7.2	Misconceptions and Fact-Checking . . . . .	70
7.2.1	Why Are Current Metrics Insufficient? . . . . .	71
7.2.2	GPT-4 as a Judge . . . . .	73
<b>8</b>	<b>Limitations of autoregressive architecture highlighted by GPT-4</b>	<b>76</b>
8.1	Warm-up with two basic examples . . . . .	76
8.2	Lack of planning in arithmetic/reasoning problems . . . . .	77
8.3	Lack of planning in text generation . . . . .	78
<b>9</b>	<b>Societal influences</b>	<b>82</b>
9.1	Challenges of erroneous generations . . . . .	82
9.2	Misinformation and manipulation . . . . .	83
9.3	Bias . . . . .	86
9.4	Human expertise, jobs, and economics . . . . .	89
9.5	Constellation of influences and considerations . . . . .	90



<b>10 Directions and Conclusions</b>	<b>92</b>
10.1 Definitions of intelligence, AI, and AGI . . . . .	92
10.2 On the path to more general artificial intelligence . . . . .	93
10.3 What is actually happening? . . . . .	94
<b>A GPT-4 has common sense grounding</b>	<b>101</b>
<b>B Appendix for multimodal and interdisciplinary composition</b>	<b>105</b>
B.1 Further details on integrative ability results . . . . .	105
B.2 Further details on vision results . . . . .	108
B.3 Graphic novel design example . . . . .	110

<b>C Appendix for the Coding section</b>	<b>111</b>
C.1 Measuring human performance on LeetCode . . . . .	111
C.2 Example of GPT-4 visualizing IMDb data. . . . .	112
C.3 More examples on visualization . . . . .	115
C.4 Example for 2D HTML game development . . . . .	116
C.5 Example for graphical user interface programming . . . . .	116
C.6 Example for reverse engineering . . . . .	119
C.7 Testing GPT-4’s ability to execute (pseudo) code . . . . .	121
<b>D Additional examples for mathematical reasoning</b>	<b>122</b>
D.1 Limitations . . . . .	122
D.2 Further examples . . . . .	126
D.3 Generating math problems with GPT-4 . . . . .	138
D.4 Mitigating calculation errors via external code execution . . . . .	139
<b>E Additional Interpretability Examples</b>	<b>141</b>
E.1 Explanation Agent Mismatches . . . . .	141
<b>F Additional examples for interaction with the world</b>	<b>144</b>
F.1 Interact with tools . . . . .	144
F.2 Examples for interaction with environments . . . . .	149



in June 2022, in an interview with Wired,  
Google computer scientist Blake Lemoine made the claims of him that

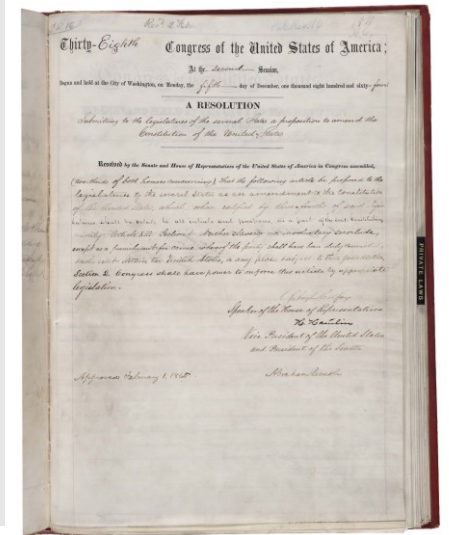
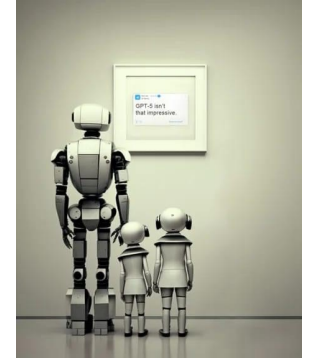
LaMDA was **sentient** and then that it is **a person**

as dictated by the Thirteenth  
Amendment to the U.S. Constitution,  
comparing it to an  
**alien intelligence of  
terrestrial origin**

non-human person?

*Section 1. Neither slavery nor involuntary servitude, except as a punishment for crime whereof the party shall have been duly convicted, shall exist within the United States, or any place subject to their jurisdiction.*

*Section 2. Congress shall have power to enforce this article by appropriate legislation*

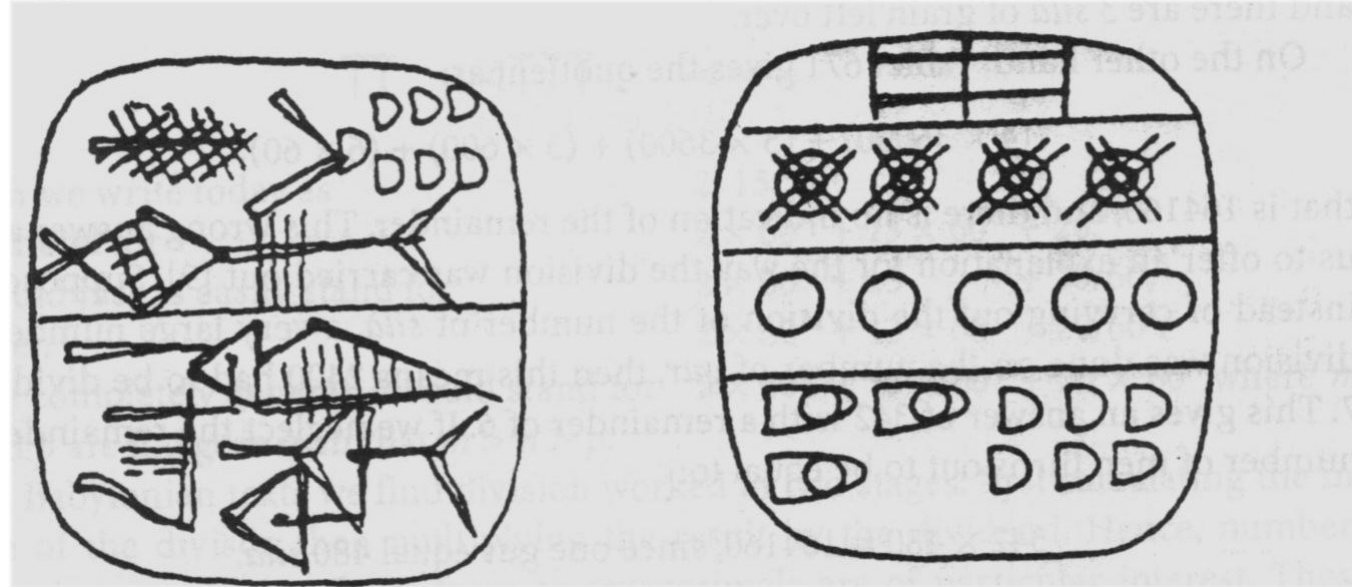


# Algorithms are ancient stuff

algorithm for dividing the grain of a barn  
among several individuals:

**input:** capacity of granary  $C$ , predetermined part  
per person  $P$

**output:** number  $N$  of people who can receive the  
portion of wheat:  $N = C/P$



Sumerian tablet, Euphrates valley, 2500 B.C.

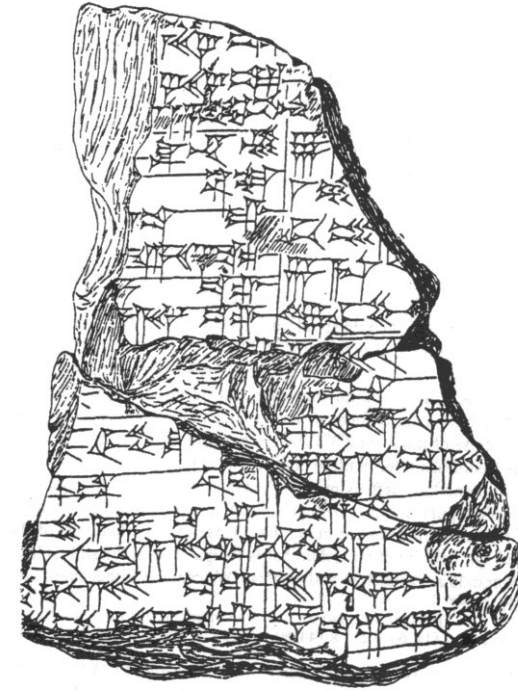
# Algorithms are ancient stuff

algorithm for dividing the grain of a barn  
among several individuals:

dividend / divisor

Step 1) calculating the inverse of the divisor

Step 2) multiplication by the dividend



Babylonian tablet, 1800 B.C.

# Algorithms are ancient stuff

A more complex algorithm: Euclid's algorithm  
for the greatest common divisor

```
def gcd(m, n) :  
    if m < n:  
        (m, n) = (n, m)  
    r = m  
    while r != 0:  
        r = m % n  
        (m, n) = (n, r)  
    return n
```



Euclid, detail of the fresco The School of Athens,  
Raphael 1510



# Algorithms are ancient stuff

A more complex algorithm: Euclid's algorithm  
for the greatest common divisor

```
def gcd(m, n) :  
    if m < n:  
        (m, n) = (n, m)  
    r = m % n  
    if r == 0:  
        return n  
    else:  
        return gcd(n, r)
```



Euclid, detail of the fresco The School of Athens,  
Raphael 1510

recursive version of Euclid's algorithm



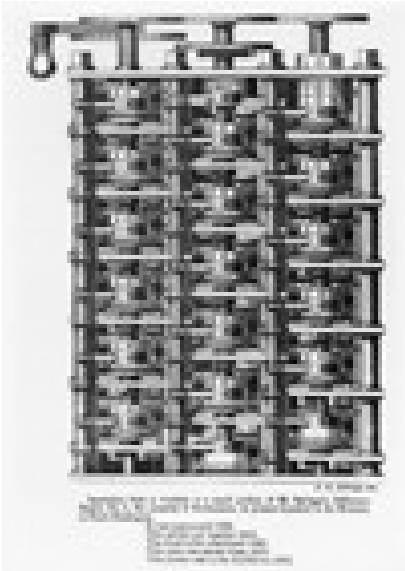
# Executing Algorithms

*«It is not worthy of excellent men to waste hours as slaves in the manual activity of calculating, which could certainly be entrusted to a machine»*

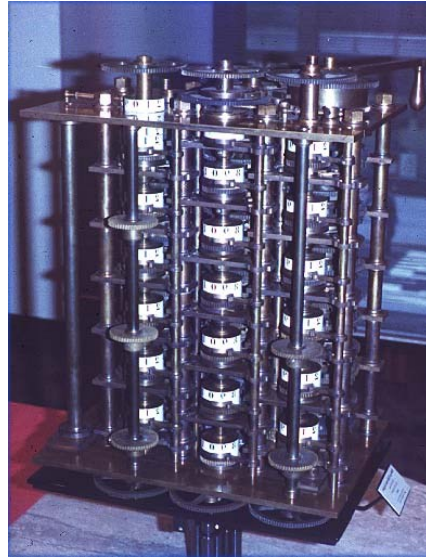


Gottfried von Leibniz  
(Leipzig 1646 -1716)

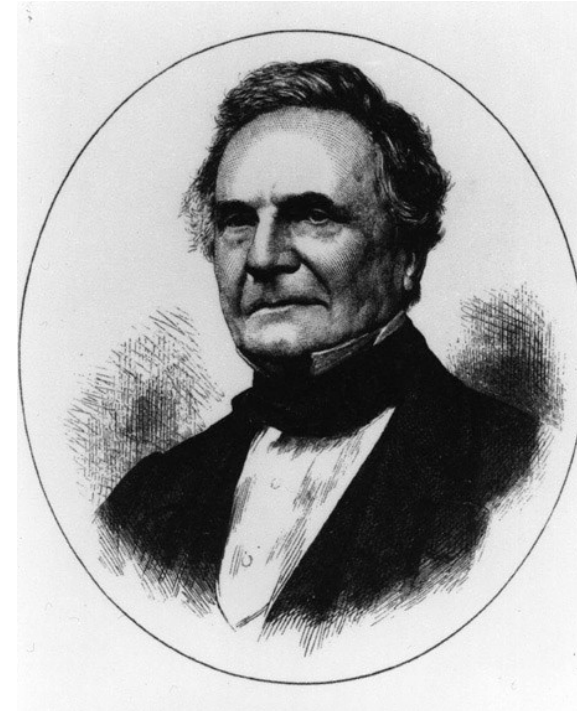
# Executing Algorithms



Difference Engine



Analytical Engine



Charles Babbage  
(London 1791-1871)

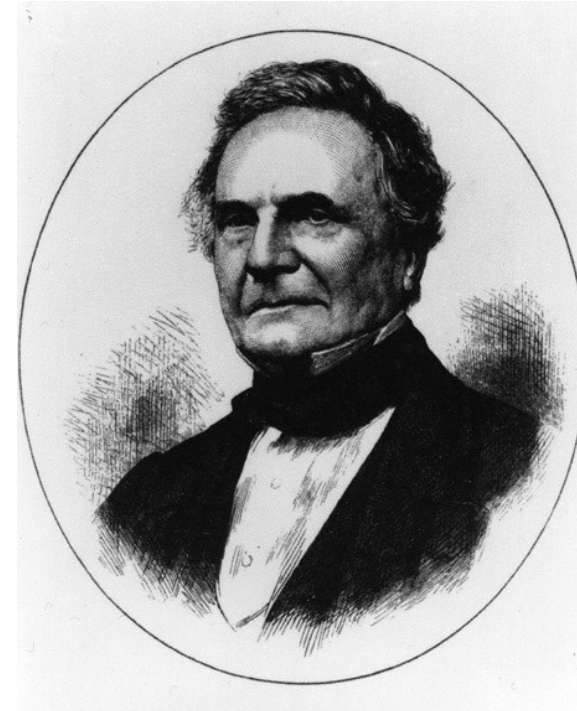
# Executing Algorithms

a politician asks Babbage :

*Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out*

Babbage's answer :

*I am not able to rightly apprehend the kind of confusion of ideas that could provoke such a question !*



Charles Babbage  
(London 1791-1871)

**Charles Babbage Institute**  
**Center for History of Information Technology**  
<http://www.cbi.umn.edu/>

Virginia Tech, USA  
**History of computing and virtual museum**  
<http://ei.cs.vt.edu/~history>

# Executing Algorithms

It is not necessary to have an infinite number of different machines available to perform different tasks. It is enough to have only one. The problems of producing various machines for different tasks turn into desk work, which consists of programming the universal machine to do those tasks. (A. Turing, 1940)

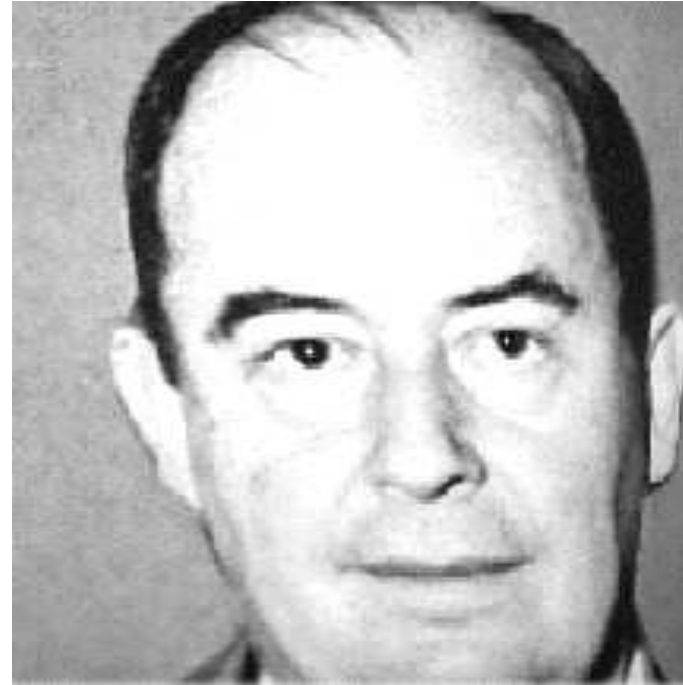
The machine must calculate, then it must contain an arithmetic central, which constitutes the first specific module.

The logical control of the machine, that is, the appropriate sequential frequency of its operations can be carried out by a central control module.

The machine must perform long sequences of operations, then it must have a considerable memory, which constitutes the third specific module.

The machine must also maintain an input - output contact with the outside.

The machine must have components to transfer information between the various modules. (J. Von Neumann, *Draft*, 1945)



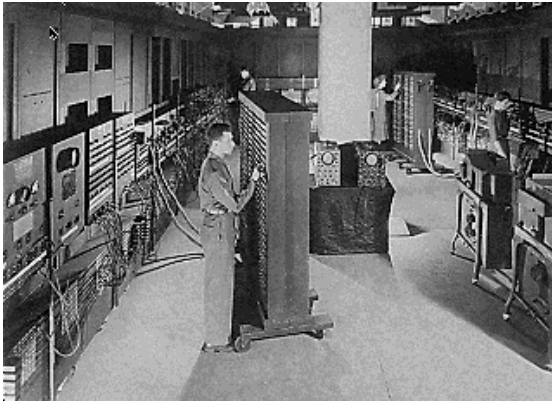
John von Neumann  
(Budapest 1903 - Washington 1957)



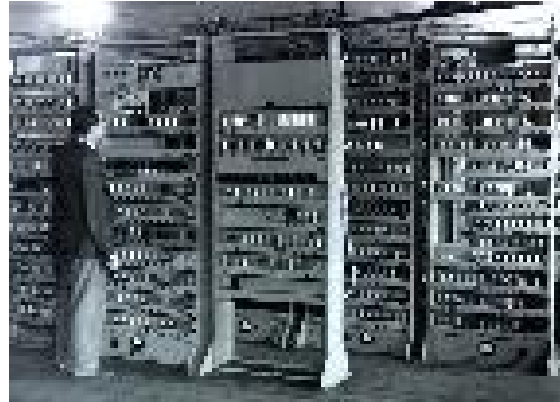
Alan Turing  
(London 1912 –  
Manchester 1954)



# Executing Algorithms



ENIAC, the first electronic calculator 1944-48



the term Computer indicated the operators

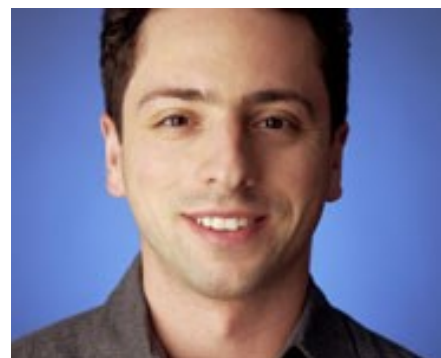
# Executing Algorithms and more



Steve Jobs e Steve Wozniak  
1977 – the **pc** is born (Apple)



Tim Berners-Lee  
1980 – the **web** is born



Larry Page e Sergey Brin  
1998 – **Google** is born



Jeff Bezos  
2003 – **Amazon** is born



Mark Zuckerberg  
2004 – **Facebook** is born



Sam Altman  
2022 – **ChatGPT**  
is born

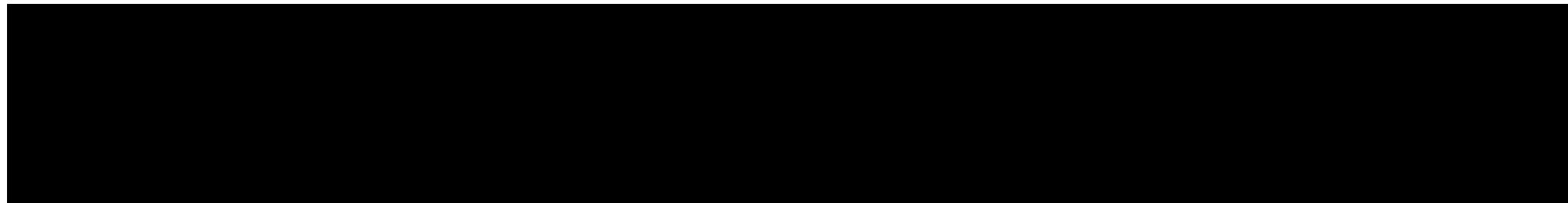
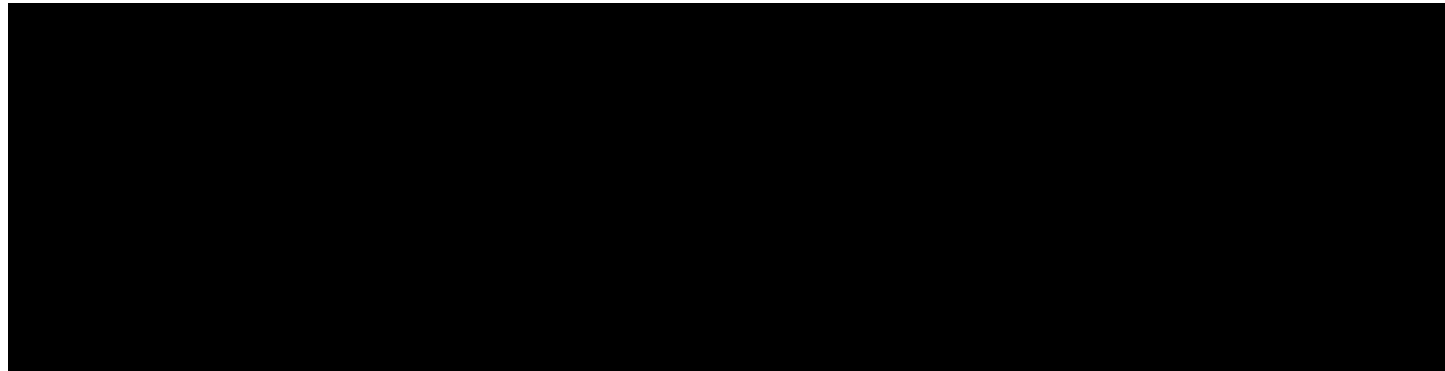
# lines of technological development

- miniaturization
- connectivity



- ✓ more computing power
- ✓ greater storage capacity
- ✓ easy access to computing power and big data availability

# Some basic terminology





# How big are big numbers ?

how big is number  $10^{20}$  ?

[www.wolframalpha.com](http://www.wolframalpha.com)

**number of people who have ever lived on Earth**

**$1.085 \times 10^{11}$  (people)**

[as estimate for the year 2017 by the Population Reference Bureau, which assumes homo sapiens appeared about 50,000 B.C.]

# How big are big numbers ?

how big is number  $10^{20}$  ?

hypothesis: average life of 60 years **convert**  
**60 years to seconds**

**$1.892 \times 10^9$  (seconds)**

hypotesis: about half of life spent talking, with a rapidity of 5 characters per second. Then, all humans who have lived on Earth have uttered altogether

**$5 \times 10^{20}$  characters;** that is 500 Exabytes allow you to memorize  
**all the words spoken by mankind**

# How good is an algorithm ?

## efficiency of an algorithm

**time** and **memory** of a computer are  
**computing resources**

the **performance** evaluation of a program is the  
quantification of the use of the computing resources  
necessary for its execution

# How good is an algorithm ?

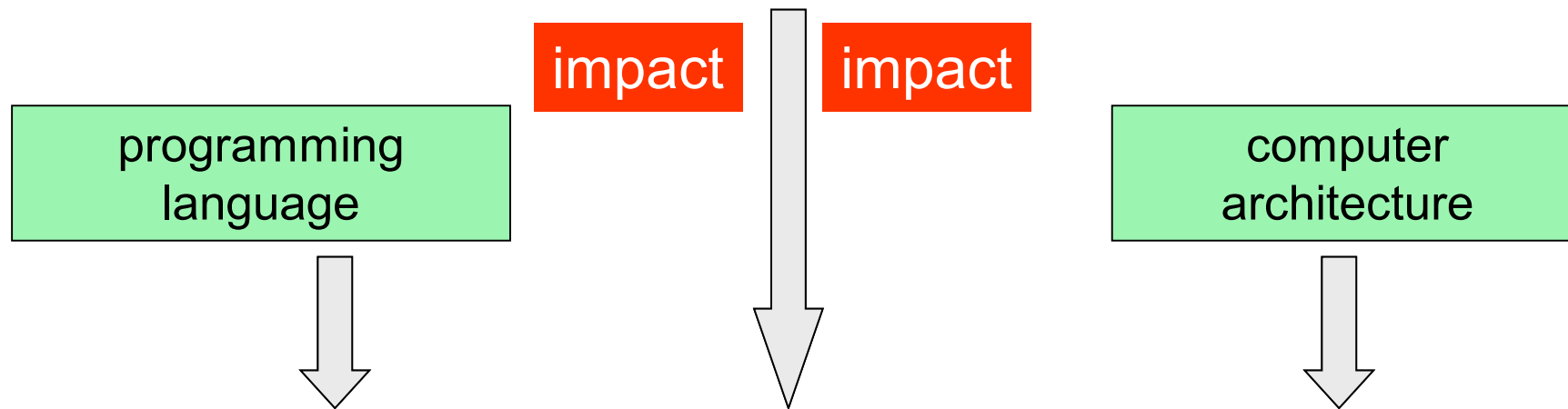
the total **number** of **operations** and **data** of an algorithm is **proportional** to the **time** and **memory** required for the execution of the program that implements the algorithm on a specific computer

## **computational complexity of an algorithm**

determine the **amount of resources** for the execution of an algorithm

# How good is an algorithm ?

**computational complexity** of an **algorithm**



**performance** of the **program (software)**  
that implements the **algorithm**

# How good is an algorithm ?

the **total number of operations** to be executed by an algorithm **depends on** the  
**number of input data**

**number of input data**

=

**computational dimension** of the (instance of the) problem

when the computational dimension of a problem grows,  
how do the number of operations (execution time) and the number of data  
(memory) of the algorithm change ?

# How good is an algorithm ?

**Computational Complexity Theory** investigates issues related to the amount of resources required for the execution of algorithms and the inherent difficulty of providing efficient algorithms to solve specific problems

- **time complexity** of an algorithm
- **space complexity** of an algorithm
- **inherent difficulty** of a **problem**

**to classify**  
**algorithms** and **problems** in **complexity classes**

# How good is an algorithm ?

## time complexity of an algorithm

- identify the **computational dimension** of the problem
- identify the **dominant operation** (or dominant operations) of the algorithm

the **time complexity function**  $T(n)$

of an algorithm is a function that expresses the number of dominant operations depending on the **computational dimension**  $n$  of the problem



# How good is an algorithm ?

## space complexity of an algorithm

- identify the **computational dimension** of the problem

the **space complexity function**  $S(n)$  of an algorithm is the function that expresses the total size of the data structures used to store input, local and output data, depending on the **computational dimension**  $n$  of the problem

# How good is an algorithm ?

**time and space complexity** of an algorithm

may also depend on  
input data values

**worst case**

**time complexity and space complexity**

# Time complexity of algorithms

```
for i in range (n):  
    q dominant operations
```

time complexity  $T(n) = qn$   
that is  
 $T(n)$  is proportional to  $n$

**LINEAR**  
time complexity

```
for i in range(n):  
    q dominant operations  
for j in range (n):  
    q dominant operations
```

time complexity  $T(n) = 2qn$   
that is  
 $T(n)$  is proportional to  $2n$   
(proportional to  $n$ )

# Time complexity of algorithms

```
for i in range (n):  
    for j in range (n):  
        q dominant operations
```

time complexity

$$T(n) = q n^2$$

that is

$T(n)$  is proportional to  $n^2$

**QUADRATIC**  
time complexity

```
for i in range (n):  
    for j in range (i,n):  
        q dominant operations
```

time complexity

$$\begin{aligned} T(n) &= q(1+2+3+\dots+n) \\ &= qn(n+1)/2 \end{aligned}$$

that is

$T(n)$  is proportional to  $n^2$

# Time complexity of algorithms

```
for i in range (n):  
    for j in range (n):  
        for k in range (n):  
            q dominant operations
```

time complexity  $T(n) = q n^3$   
that is  
 $T(n)$  is proportional to  $n^3$

**CUBIC**  
time complexity

# Time complexity of algorithms

```
i = 0  
while 2**i < n:  
    i = i+1  
    q dominant operations
```

time complexity  $T(n) = q \log_2 n$   
that is  
 $T(n)$  is proportional to  $\log_2 n$

**LOGARITHMIC**  
time complexity

# Time complexity of algorithms

how many times can you divide an array of size  $n$  in half?



$n=8$



**LOGARITHMIC**  
time complexity

$n$  power of 2:  $\log_2(n)$  times

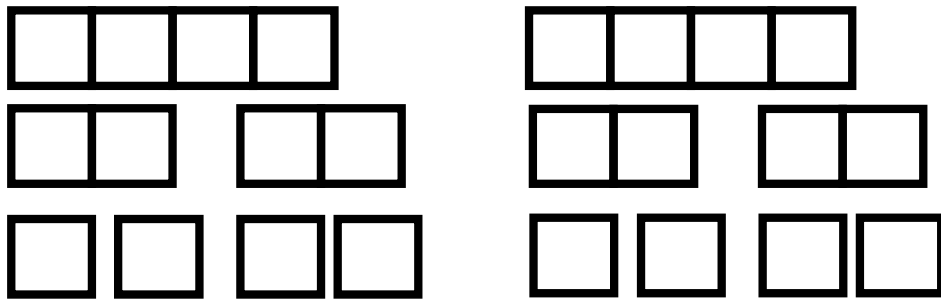
in general: smallest integer greater than  $\log_2(n)$  times

# Time complexity of algorithms

algorithms that carry out  $\log_2(n)$  steps and at each step execute  $n$  dominant operations



$n = 8$



time complexity  $T(n) = n \log_2 n$   
that is  
 $T(n)$  is proportional to  $n \log_2 n$

**LIN-LOG**

time complexity

$n$  power of 2:  $n \log_2(n)$  operations



# Asymptotic time complexity of algorithms

Let  $f(n)$  and  $g(n)$  be two **non negative** and **non decreasing** functions, then

$$f(n) = O(g(n))$$

$f$  is of the order of  $g$

if there are two **positive constants**  $c$  and  $N_0$  such that:

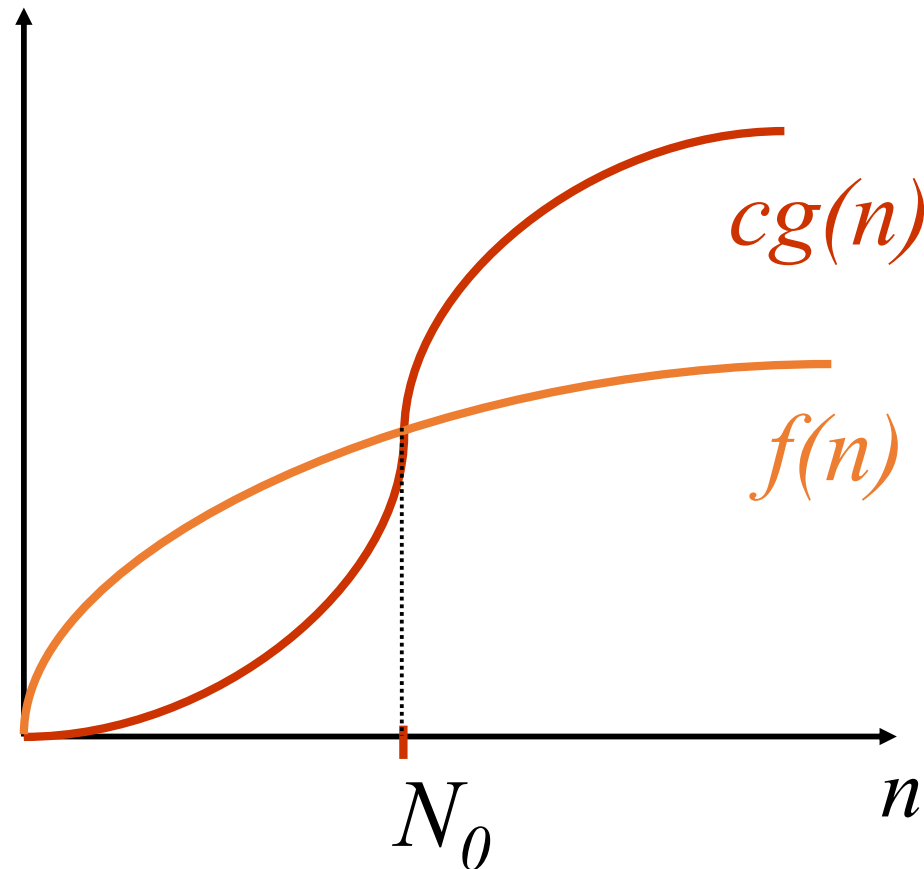
$$f(n) \leq cg(n)$$

$$n \geq N_0$$

# Asymptotic time complexity of algorithms

$$f(n) = O(g(n))$$

the curve  $f(n)$  stays **below**  
(or coincides) the curve  $cg(n)$ ,  
starting from  $N_0$



# Asymptotic time complexity of algorithms

$$f(n) = O(g(n))$$

Exercise: if  $f(n) = \sum_{i=0}^p a_i n^i$

that is,  $f(n)$  is a polynomial of degree  $p$  then:  $f(n) = O(n^p)$

Exercise: if  $f(n) = 2n^2 + 3n + 5$  then:  $f(n) = O(n^2)$

# Asymptotic time complexity of algorithms

$$f(n) = O(g(n))$$

Exercise: give a function  $f(n)$  such that

$$f(n) = O(1)$$

Exercise: yes or no:

$$n = O(n^2)$$

# Complexity classes

$$T(n) = O(1)$$

constant

$$T(n) = O(\log_2 n)$$

logarithmic

$$T(n) = O(n)$$

linear

$$T(n) = O(n \log_2 n)$$

lin-log

$$T(n) = O(n^2)$$

quadratic

$$T(n) = O(n^k)$$

polynomial of degree  $k$

$$T(n) = O(2^n)$$

exponential

$$T(n) = O(n!)$$

factorial

# Can we always design an algorithm to solve any problem?

complexity	1 sec	1 minute	1 hour
$O(n)$	$10^8$	$6 \cdot 10^9$	$3.6 \cdot 10^{11}$
$O(n \log_2 n)$	$\approx 4 \cdot 10^6$	$\approx 2 \cdot 10^8$	$\approx 1 \cdot 10^{10}$
$O(n^2)$	$10^4$	77459	$6 \cdot 10^5$
$O(2^n)$	26	32	38
$O(n!)$	11	12	14

dimensions of problems that can be solved, with a **100 Mops/sec computer**

in WolframAlpha: `solve 10^(-8) * x^2 = 60`

# Can we always design an algorithm to solve any problem?

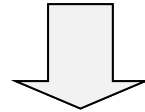
complexity	1 sec	1 minute	1 hour
$O(n)$	$10^{12}$	$6 \cdot 10^{13}$	$3.6 \cdot 10^{15}$
$O(n \log_2 n)$	$2.5 \cdot 10^{10}$	$\approx 1.4 \cdot 10^{12}$	$\approx 7.5 \cdot 10^{13}$
$O(n^2)$	$10^6$	$\approx 7.7 \cdot 10^6$	$6 \cdot 10^7$
$O(2^n)$	35	45	51
$O(n!)$	14	16	17

dimensions of  
problems that can be  
solved,  
with a **1000000**  
**Mops/sec computer**

in WolframAlpha: `solve 10^(-12) * 2^x = 60`

# Can we always design an algorithm to solve any problem?

algorithms that can be used for effective problem solving



**polynomial** time complexity algorithms

~~exponential/factorial time complexity algorithms  
cannot be used  
in any practical problem solving~~



# Can we always design an algorithm to solve any problem?

exponential/factorial time complexity algorithms **cannot be used**  
in any practical problem solving

- ✓ algorithm of **factorial** time complexity,
- ✓ computer which executes 1 operazione in  $10^{-12}$  sec,
- ✓ problem of computational dimensione 100

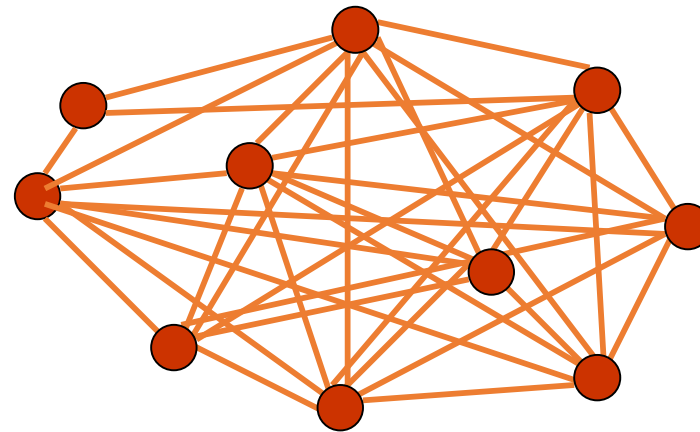
in WolframAlpha:

```
10(-12) *factorial(100) / (86400*365)
```


total execution time :  $10^{138}$  years

# Are there problems solved only by exponential/factorial time complexity algorithms ?

TSP, *travelling salesman problem*



 city

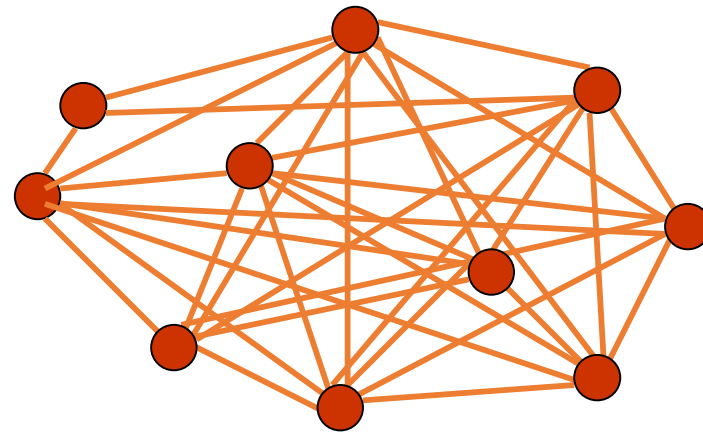
 road

$$T(n) = O(n!)$$


given a set of cities and travel costs, from any city to any other city, determine the **cheapest** route that allows you to visit **each** city **exactly once** and then return to the city of departure

# Are there problems solved only by exponential/factorial time complexity algorithms ?

TSP, *travelling salesman problem*



 city

 road

$$T(n) = O(n!)$$

The only known algorithm that solves this problem is a brute force algorithm that computes all possible routes and determine the cheapest one

The number of possible routes is equal to the number of the permutations of  $n$  cities

# Algorithms ARE technology

problem of  
computational  
dimension  $n=10^7$

computer power

execution time

algorithm of  
quadratic complexity

10000 Mops/sec

$(10^7)^2$  operations  

---

 $10^{10}$  operations/sec

$=10^4$  sec

algorithm of  
lin-log complexity

100 Mops/sec

$10^7 \cdot \log_2 10^7$   

---

 $10^8$  operations/sec

$=2$  sec



MASTER IN ENTREPRENEURSHIP  
INNOVATION MANAGEMENT  
IN COLLABORATION WITH **MIT SLOAN**

IN COLLABORATION WITH

**MIT** MANAGEMENT  
SLOAN SCHOOL



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**PARthenope**

MASTER MEIM 2022-2023

Enjoy our courses on Artificial intelligence, Digital  
Strategies and Digital Technology

..... to be continued