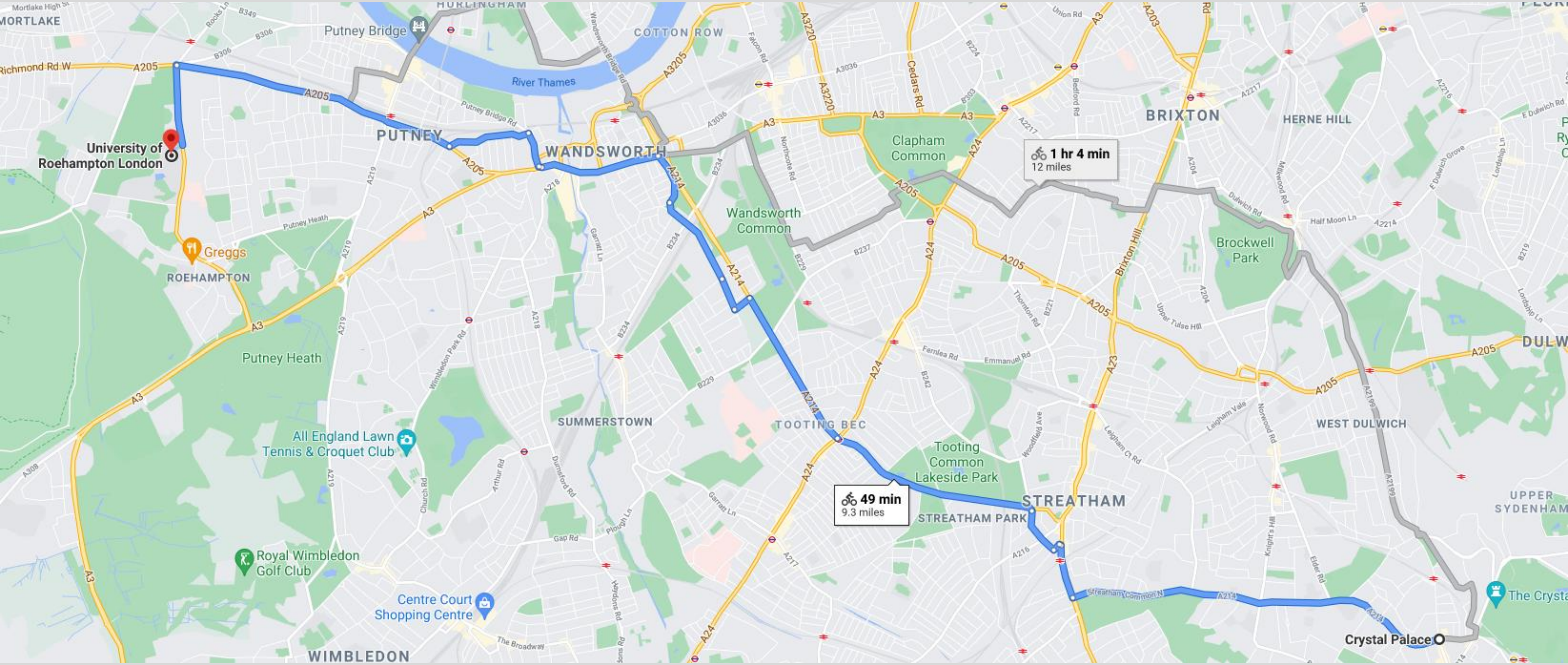
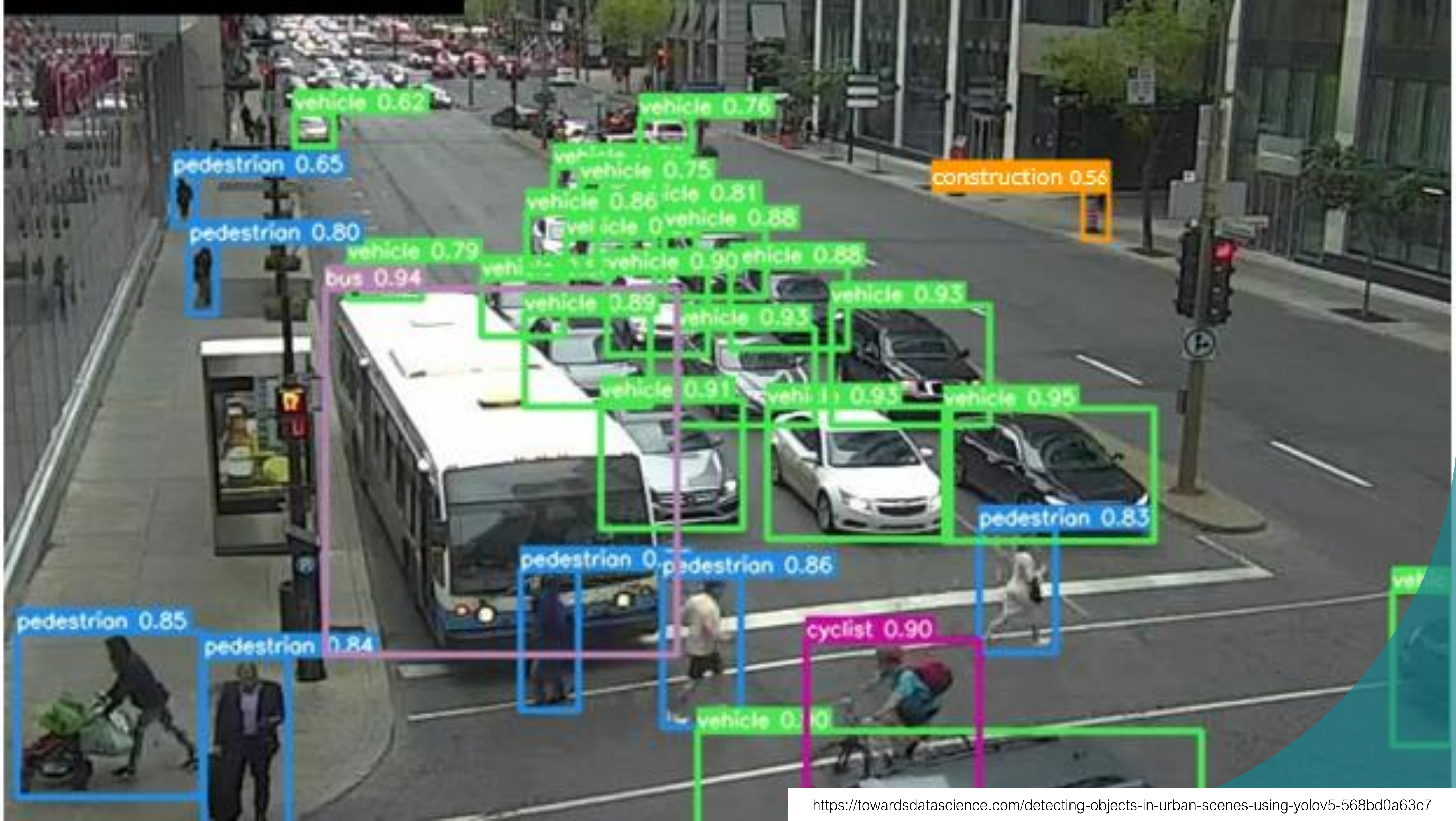
University of Roehampton London

Algorithms

BSC COMPUTER SCIENCES – YEAR 2

DR ROBERTO MURCIO


University of Roehampton London
Crystal Palace
49 min
9.3 miles
1 hr 4 min
12 miles
PUTNEY
WANDSWORTH
ROEHAMPTON
TOOTING BEC
STREATHAM
STREATHAM PARK
BRIXTON
HERNE HILL
WEST DULWICH
SUMMERSTOWN
WIMBLEDON
MORTLAKE
COTTON ROW
HURLINGHAM
UPPER SYDENHAM
River Thames
Putney Bridge
Clapham Common
Wandsworth Common
Tooting Common Lakeside Park
Brockwell Park
Putney Heath
All England Lawn Tennis & Croquet Club
Royal Wimbledon Golf Club
Centre Court Shopping Centre
Greggs
The Crysta

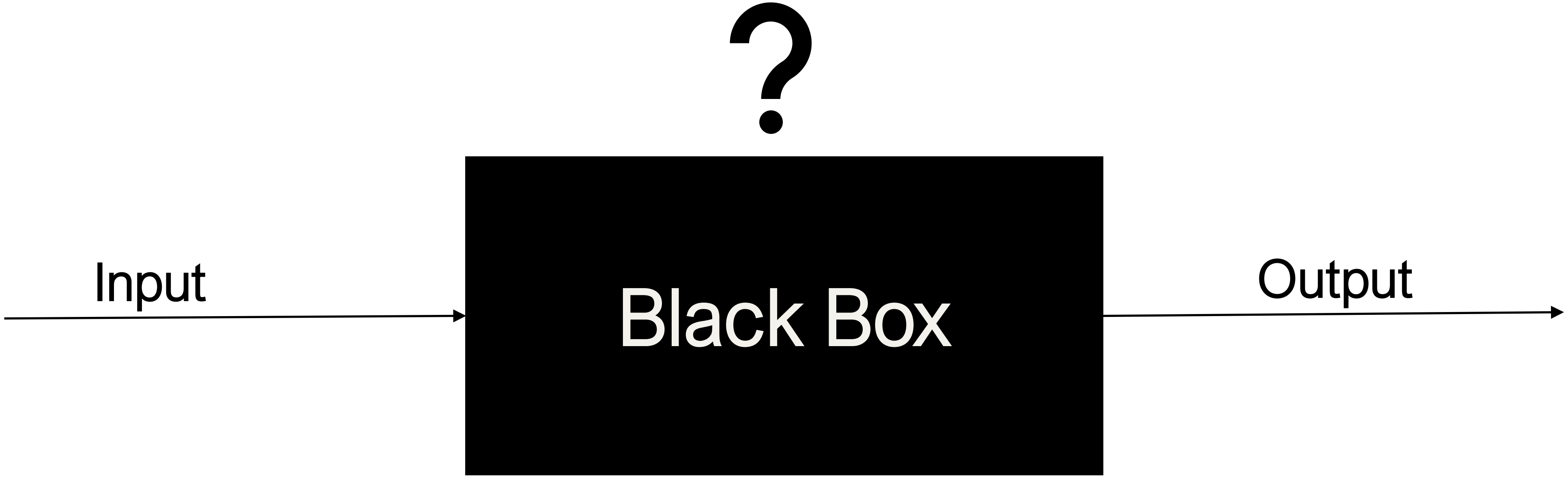


https://towardsdatascience.com/detecting-objects-in-urban-scenes-using-yolov5-568bd0a63c7

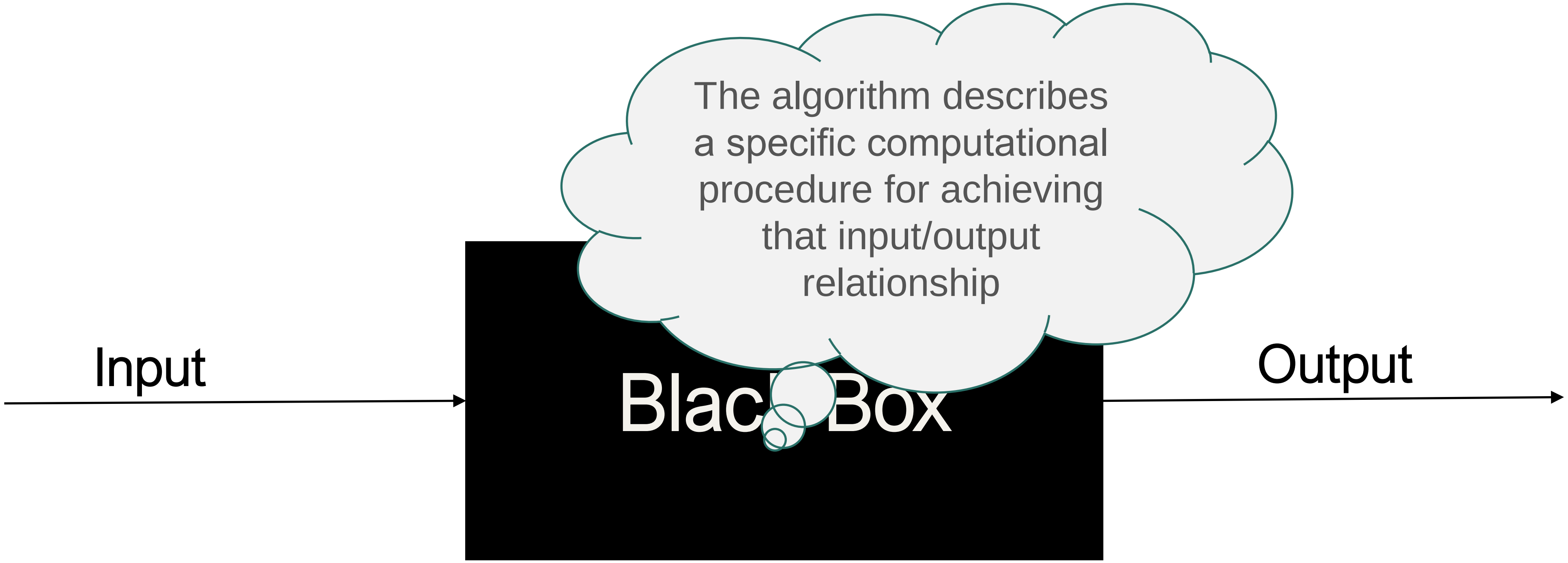
Introduction

What is an algorithm?

Any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output [Introduction to Algorithms, Cormen]



What is an algorithm?



The algorithm describes a specific computational procedure for achieving that input/output relationship

What is an algorithm?

Example - Sorting

Input: A sequence of numbers $\langle a_1, a_2, \dots, a_n \rangle$

Output: A permutation (new arrange) $\langle a'_1, a'_2, \dots a'_n \rangle$ of the input sequence such that $a'_1 \le a'_2 \le \cdots \le a'_n$

Correctness

Your algorithm should produce the expected answer for every input.

Your algorithm need to finish (halt) in a finite amount of time.

Correctness of sorting

If for every $\langle a_1, a_2, \dots, a_n \rangle$ *halts* and produce $\langle a'_1, a'_2, \dots a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$

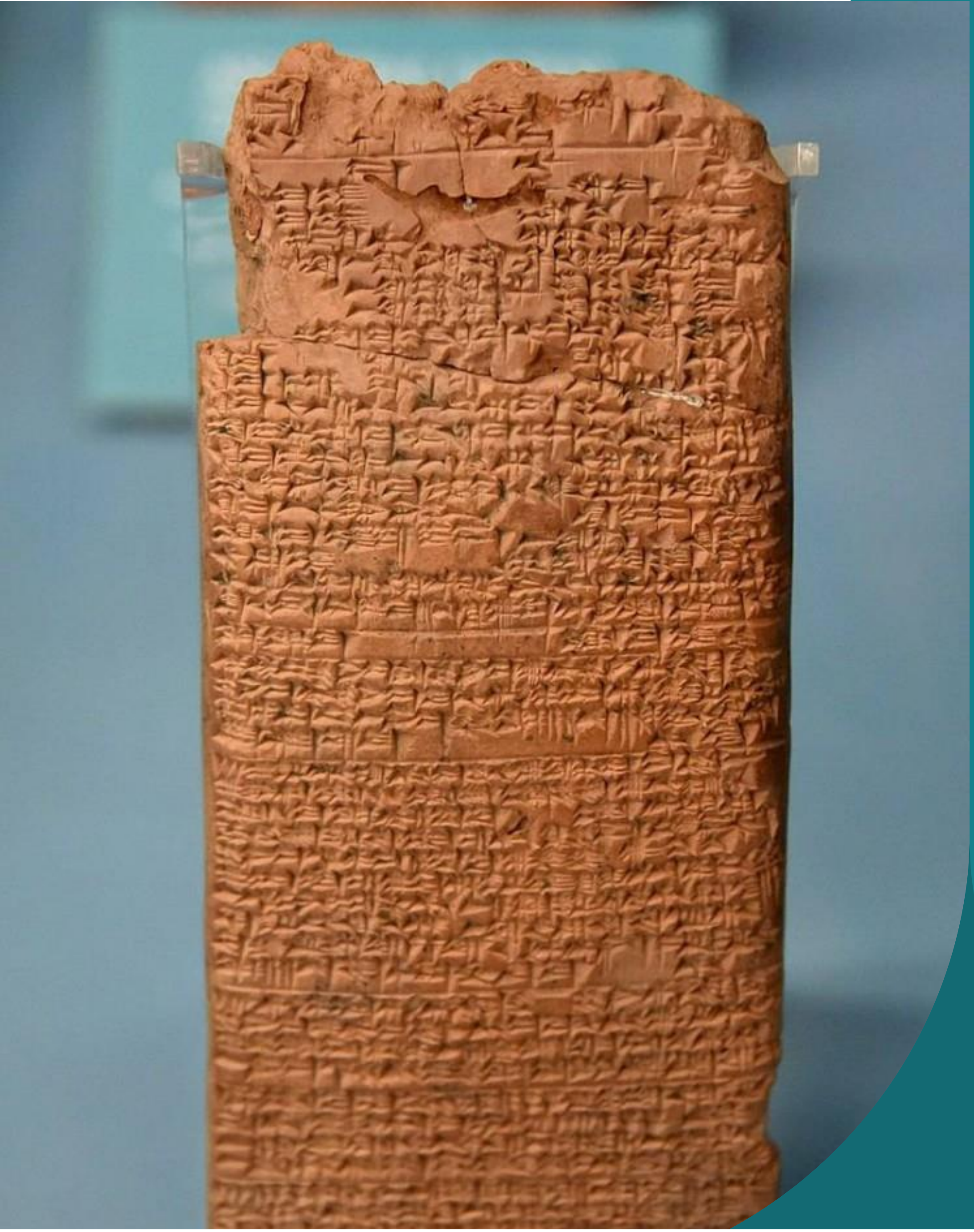
we say that the sorting algorithm is correct.

Academic examples

Greater common divisor (GCD)

Matrix multiplication

Fibonacci numbers



An 18th Century BC medical tablet, Source: Osama Shukir Muhammed Amin FRCP/Wikimedia Commons

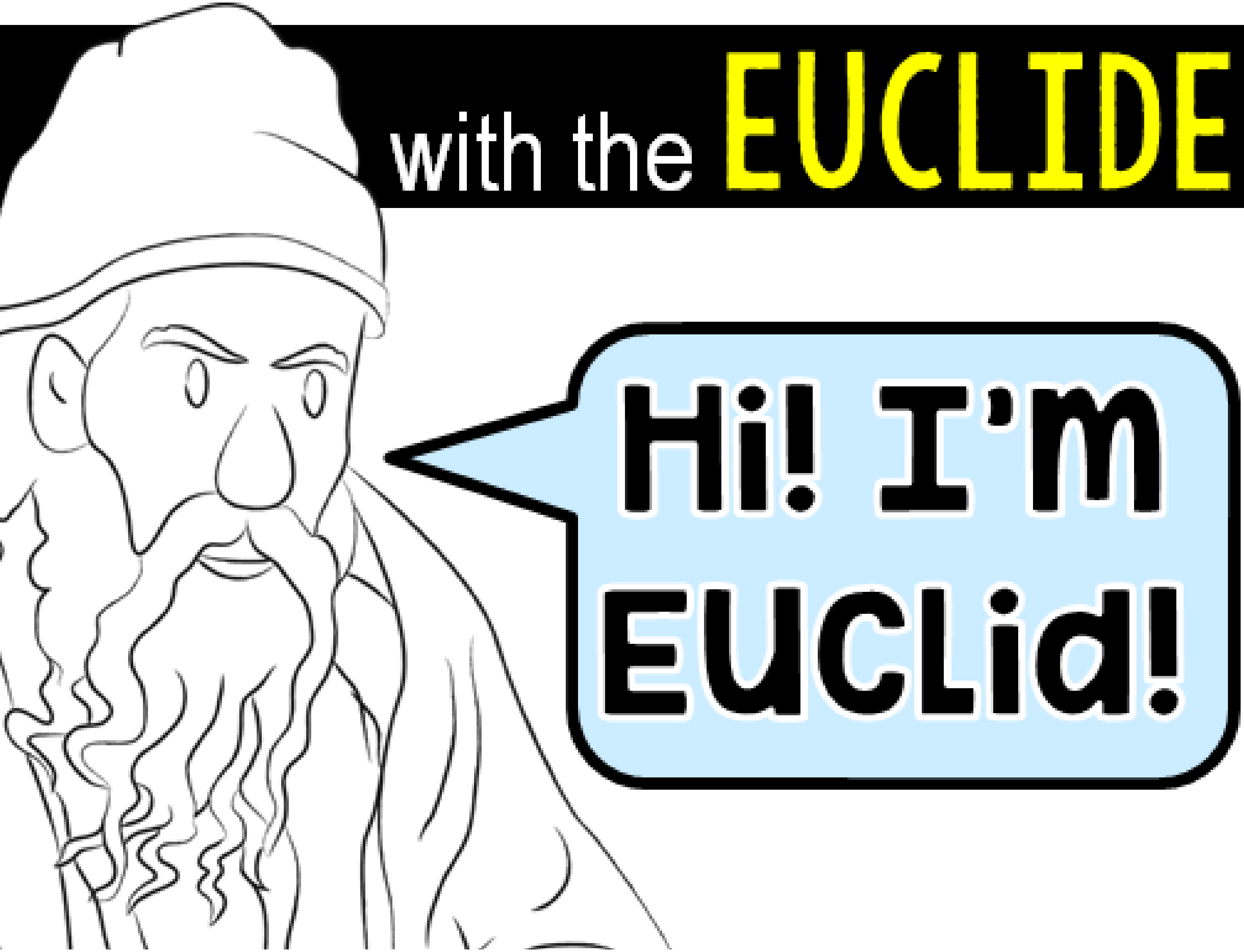
First Algorithms

- "The Babylonian mathematicians were not limited simply to the processes of addition, subtraction, multiplication, and division; they were adept at solving many types of algebraic equations”

Donald E. Knuth "Ancient Babylonian Algorithms

Greatest Common Factors

with the EUCLIDEAN ALGORITHM



$81 = 1(57) + 24$

$57 = 2(24) + 9$

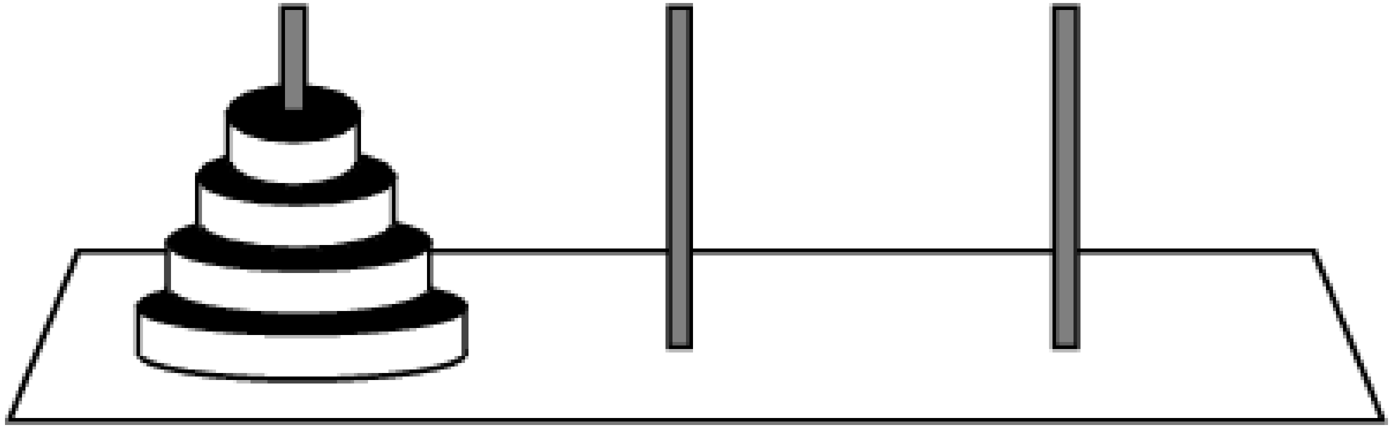
$24 = 2(9) + 6$

$9 = 1(6) + 3$

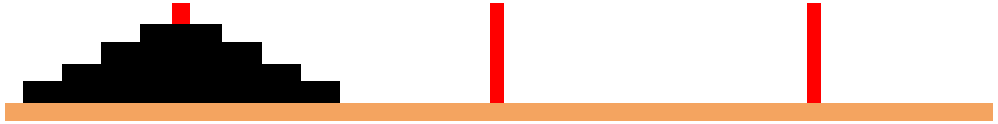
$6 = 2(3) + 0$. stop

https://www.scaffoldedmath.com/2018/02/euclidean-algorithm.html

Towers of Hanoi
(E. Lucas, 1883)



- Given a stack of disks arranged from largest on the bottom to smallest on top placed on a rod, together with two empty rods, your task is to move the stack from one rod to another.
- You can only move one disk at a time and,
- You can not place a larger disks on top of a smaller disk



Towers of Hanoi

- Two disks case → d1 to C; d2 to B; d1 to B → 3 moves
- Three disks case → d1 to B; d2 to C; d1 to C; d3 to B; d1 to A; d2 to B; d1 to B → 7 moves
- Four disks case → Solve for three disks (7 moves) + d4 to C; d1 to C; d2 to A; d1 to A; d3 to C; d1 to B; d2 to C; d1 to C → 15 moves

N disks	# Moves
2	3
3	7
4	15

5	31

Towers of Hanoi

- Move the top N−1 disks to an intermediate rod.
- Move the bottom disk to the destination rod.
- Finally, move the N−1 disks from the intermediate peg to the destination peg.
- $a_1 = 1; a_2 = 3;$

$$a_n = a_{n-1} + 1 = 2^n - 1$$

Why?

Some Real life examples

Google's PageRank (ranking)

Facebook suggestions

Barclays Fraud detection

Efficiency

We do not only want our algorithms to be correct, but we want them to be efficient.

We want them to run in the minimum number of steps!

Efficiency

Number of steps != Real execution time

Execution time is Hardware related

Tower of Hanoi takes $2^n - 1$ steps, but how many seconds take each step?

Efficiency - Sort Example

Insertion sort $c_1 n^2$ steps

Merge sort $c_2 nlogn$ steps

Which one is faster?

Efficiency - Sort Example

N	Insert	Merge
10	2*10^2 = 200	50*10*lg(10)= 500
100	2*100^2 = 20,000	50*100*lg(100)= 10,000
1000	2*1000^2 = 2,000,000	50*1000*lg(1000)= 150,000
10000	2*10000^2 = 200,000,000	50*10000*lg(10000)= 2,000,000

c1 = 2; c2=50

Efficiency - Sort Example

But what about Hardware

Computer A running insertion sort - 10 billion instructions per second

Computer B running merge sort - 10 million instructions per second

Computer A is 1000 times faster than computer B in raw computing power.

They each must sort an array of 10 million numbers.

Again, c1 = 2; c2=50 (insertion was coded “better”)

Efficiency - Sort Example

computer A $$\frac{2\cdot\left(10^7\right)^2\ instructions}{10^{10}\ instructions/second} = 20{,}000\ secs$$

computer B $$\frac{50\cdot 10^7 lg 10^7 instructions}{10^7\ instructions/second} \approx 1{,}163\ secs$$

Summary

- Modern technologies are based on a series of different algorithms

- An algorithm is a sequence of steps that produces a **correct** answer for every input and finishes in a finite time **(halt)**

- Algorithms has been with us for a long time

- We want our algorithms to be efficient

Thanks!