

Data Structures and Algorithms

Introduction



Who are we?



Course Coordinator

Tom Pepels

tom.pepels@maastrichtuniversity.nl

Teachers



Francesco Barile, PhD



Bastian Küppers, PhD



Spriha Joshi, MSc

{f.barile,b.kuppers,spriha.joshi}@maastrichtuniversity.nl



Maastricht University

Introduction

- Data Structures and Algorithms?
 - Why should I care?
- What will we learn?
 - How?
- How am I graded?
- Logistics – lectures, exam



Data Structures and Algorithms



Data structures?

Data structures?

- How to organize notes?



Data structures?

- How to organize notes?



- How to organize people?

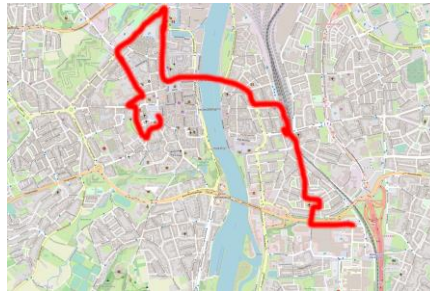


Data Structures?

- The “**best**” ways to **represent data** in software in relation to a specific task
 - What is data?
 - What does the “best” mean?
 - Why do I need to “represent” data?

Data Structures?

- The “**best**” ways to **represent data** in software in relation to a specific task
 - What is data?
 - What does the “best” mean?
 - Why do I need to “represent” data?



Algorithms?

- Solutions to solve a problem in the **best** way
- So far, we cared about being **correct**
- Now we are going look into how being **efficient**
 - How slow/fast my algorithm is
 - What are the memory requirements



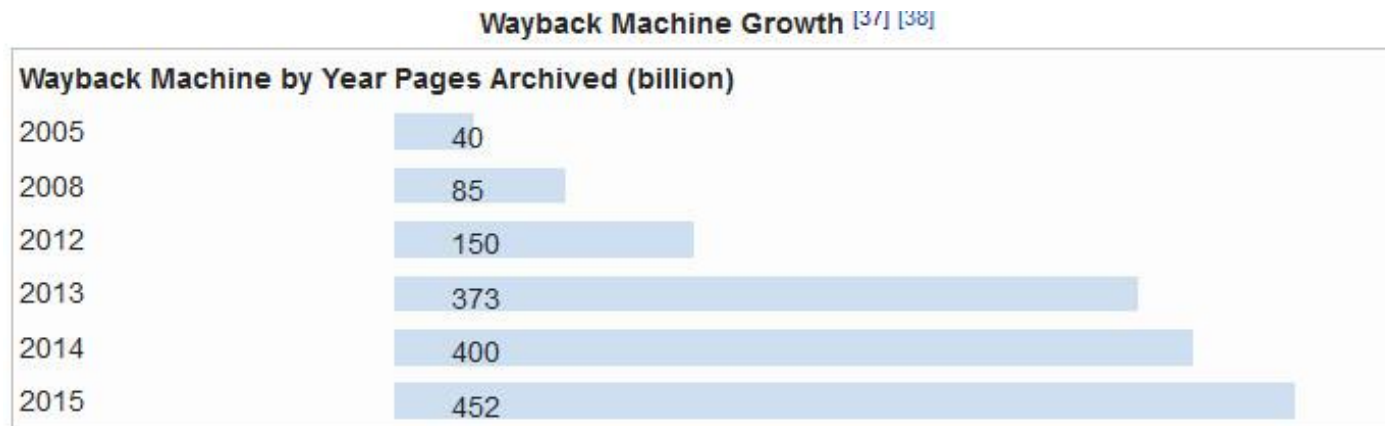
Motivation

- You want to be fast



Motivation

- You want to be fast
- You have to be fast
 - Data Science on web-size data
 - Wayback Machine of archive.org contains half a trillion websites
 - Google translate:
 - Translate more than 100 billion words per day



More motivation: Application – Job interviews

- High technology companies tend to ask questions about algorithms and data structures during job interviews.
- Algorithms questions can be short but often require critical thinking, creative insights, and subject knowledge.
- All the “Applications” exercises in Chapter 1 of the Goodrich-Tamassia textbook are taken from reports of actual job interview questions.
- Example:
 - Implement a queue using two stacks

Course Overview

Course overview

- Systematic design and application of data structures and algorithms.
 - Algorithms and their complexity
 - Concept of Abstract Data Types (ADT)
 - List, Queue, Stack, Set, Tree, Graph
 - Algorithm design principle

Course overview

- Course content is based on Java
 - Knowledge of Object Oriented programming is required!
<https://docs.oracle.com/javase/tutorial/java/concepts/>
- But theory can be applied to (almost) all programming languages
 - C++
 - C#
 - Python / Ruby

Course structure

Course combines theory and practice

- Theory: Inverted classroom
 - Weekly online contents (videos + books chapters)
 - Quizzes on Canvas to practice
 - Weekly lectures (live coding / problem solving)
 - Weekly Q&A session **(optional)**
 - Catch up lecture (week 3 and 6) **(optional)**
- Practice: tutorials
 - Weekly sessions (small groups with a TA)
 - Debug sessions **(optional)**



Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)				



Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)				



- > Watch Videos
- > Read Book
- > 1st try quiz

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)		



- > Watch Videos
- > Read Book
- > 1st try quiz

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)		



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on wooclap!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)		



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on wooclap!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on wooclap!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodlap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodlap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Debug code with TAs



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Debug code with TAs



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional) + Catch up (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Debug code with TAs



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional) + Catch up (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Debug code with TAs



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Week 3 and 6
- > Lecture last 3 weeks topics



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Week plan

		Monday	Tuesday	Wednesday	Thursday	Friday
		Preparation (essential) + Debugging (optional)	Lecture (essential)	Lecture (essential)	Q&A (optional) + Catch up (optional)	Tutorial (essential) + Implement (essential)



- > Watch Videos
- > Read Book
- > 1st try quiz



- > Debug code with TAs



- > Tuesday and/or Wednesday
- > Live coding
- > Additional explanation of some topics
- > You decide the topics on woodclap!



- > Come and ask questions on any topic still not clear
- > (ideally) last try quiz



- > Week 3 and 6
- > Lecture last 3 weeks topics



- > Work on a task in small groups with a TA
- > Produce a pseudo-code



- > Implement your solutions!

Scheduling

		Monday	Tuesday	Wednesday	Thursday	Friday
W1	5/2		Lecture w1.1	Lecture w1.2	Q&A w1	Tutorial w1
CARNIVAL WEEK						
W2	19/2	Debug w2	Lecture w2		Q&A w2	Tutorial w2
W3	26/2	Debug w3	Lecture w3	Lecture w3	Catch up w3	Tutorial w3
W4	4/3	Debug w4	Lecture w4		Q&A w2	Tutorial w4
W5	11/3	Debug w5	Lecture w5		Q&A w2	Tutorial w5
PROJECTS WEEK						
W6	25/3	Debug w6		Lecture w6	Catch up w6 Q&A w6	

Scheduling

		Monday	Tuesday	Wednesday	Thursday	Friday
W1	5/2		Lecture w1.1	Lecture w1.2	Q&A w1	Tutorial w1
CARNIVAL WEEK						
W2	19/2	Debug w2	Lecture w2		Q&A w2	Tutorial w2
W3	26/2	Debug w3	Lecture w3	Lecture w3	Catch up w3	Tutorial w3
W4	4/3	Debug w4	Lecture w4		Q&A w2	Tutorial w4
W5	11/3	Debug w5	Lecture w5		Q&A w2	Tutorial w5
PROJECTS WEEK						
W6	25/3	Debug w6		Lecture w6	Catch up w6 Q&A w6	



***“Keep an eye on the timetables.
They like to change” [semi-cit.]***

Grading

Exam grade + Bonus points

- No graded Assignments
- No mandatory attendance

BUT

- 0.5 bonus percentage if you pass all quizzes
 - at least 60% grade on each quiz
- 0.5 bonus percentage if you attend 4 out of 5 tutorials
 - Attendance will be checked by the TAs during tutorials

Grading

- Given your eXam percentage grade $x\%$
- Your Bonus points p

Your bonus is $x\%$ of p

- Exam grade: 65%, Bonus: 0.5 points
 - Calculation: $6.5 + (0.65 * 0.5) = 6.825$ - Final grade: 7
- Exam grade: 40%, Bonus: 1 point
 - Calculation: $4 + (0.4 * 1) = 4.4$ - Final grade: 4
- Exam grade: 80%, Bonus: 1 point
 - Calculation: $8 + (0.8 * 1) = 8.8$ - Final grade: 9

Exam

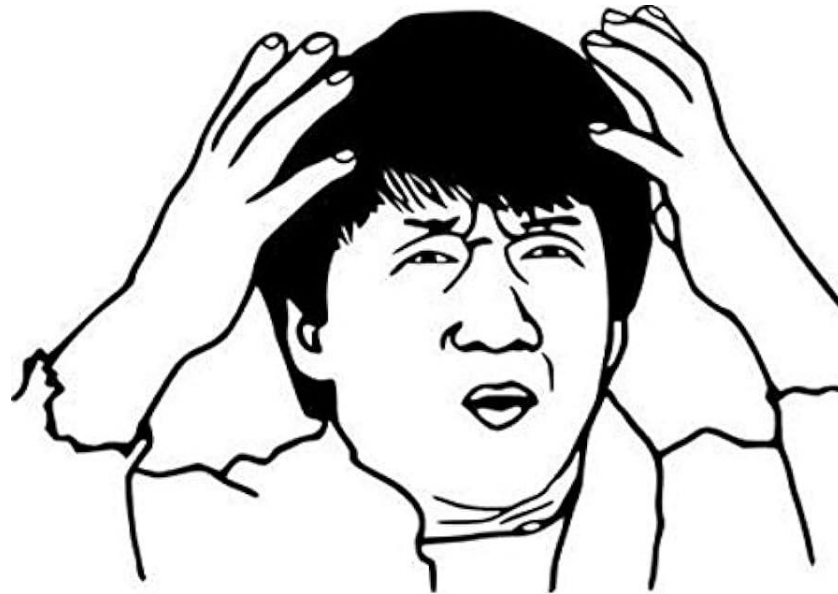
- Exam consists of 3 questions of type A and 2 questions of type B
 - A questions are more theoretical - *similar to the theoretical assignments (quizzes)*
 - B questions asks you to define an algorithm for performing a task - *similar to tutorial exercises*
 - You choose 2 questions of type A and 1 question of type B
- Let's have a look at last year's exam

Exam

- This year we use a new hybrid format
 - 1h A questions on paper
 - 1h B questions

Exam

- This year we use a new hybrid format
 - 1h A questions on paper
 - 1h B questions - using your laptop!



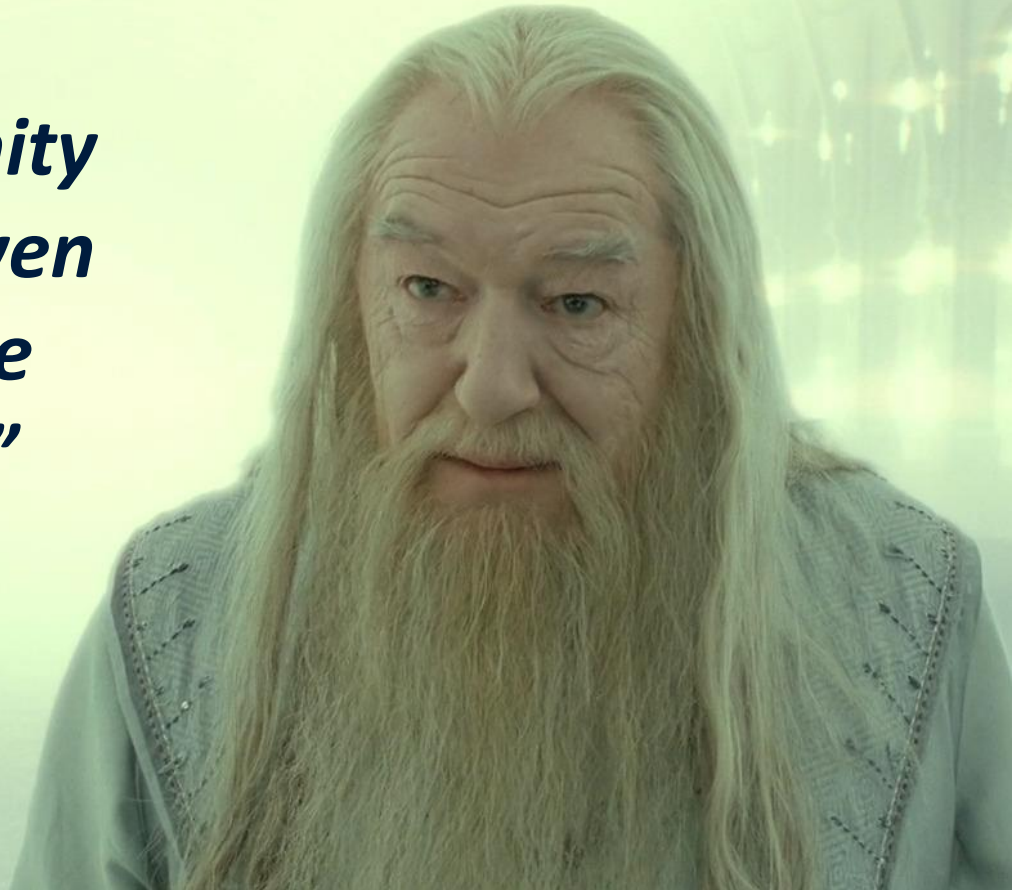
Exam

- This year we use a new hybrid format
 - 1h A questions on paper
 - 1h B questions - using your laptop!
- We will ask you to install a software on your laptop **just for the time of the exam**
 - Please check the requirements and email us if you have any problems:
<https://www.schoolyear.com/frequently-asked-questions>
 - We will have a test session in week 3 or 4

What if I fail?

What if I fail?

***“A Resit opportunity
will always be given
at DACS to those
who deserve it”
[semi-cit]***



What if I fail?

- If you have a failing overall grade (**lower than 6**) you can take a Resit exam
- The bonus will then be recomputed based on your Resit exam grade

Textbook

- Sedgewick and Wayne (2011)
Algorithms Fourth Edition.
Addison Wesley.
ISBN: 978-0321573513
(Mandatory)
- A Y Bhargava (2016).
Grokking Algorithms: An Illustrated Guide
for Programmers and Other Curious People.
Manning.
ISBN: 978-1617292231
(Additional but recommended)

