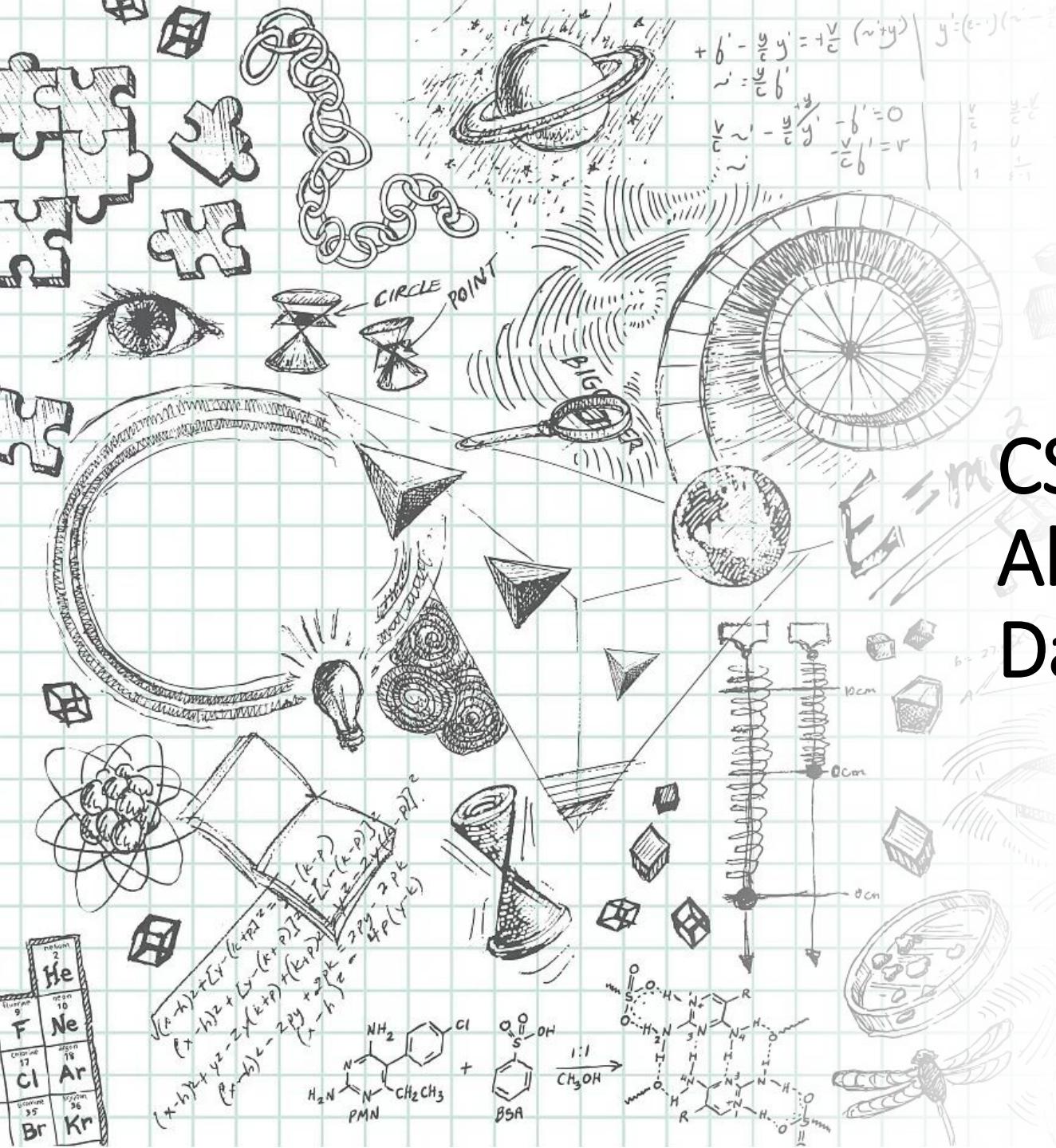


CS210 – Algorithms and Data Structure



Topic 0 – About this Course



Outline

- **Classes & Labs**
- Course content
- Grading
- Tools to be used in this course
- Motivation

Classes & Labs

- Course lecturer:
 - Guangyuan Piao (guangyuan.piao@mu.ie)
 - TA: Dongqu Han
- Lectures:
 - Time (CSSE): Mon, 15:50 – 17:30 (CN)/8:50 – 10:30 (IE)
 - Time (RIDS&MMWD): Mon, 19:00 – 20:40 (CN)/12:00 – 13:40 (IE)
 - Location: Online
- Labs:
 - Time: Sun, 14:00 – 17:30, even weeks from 4th to 18th (x8)
 - Location: will update on Moodle

Connectivity feedback



Send me a message during the class in case I lost network connectivity



Send me feedback for issues regarding classroom

network connectivity

audio quality

...

Outline



- Classes & Labs
- **Course content**
- Grading
- Tools to be used in this course
- Motivation



Course Content

- This class is **NOT** a programming course
 - But you need the basic ability of Java programming.
- The idea is to provide you with concepts: tools you need to write efficient programs
- The programming language you use and the way you implement these ideas is up to you
- You will learn efficient structures for storing information and handy algorithms for manipulating that information

What the course is about

- There are four main, interrelated topics in **this class**:
 - Recursion
 - Data structures
 - Algorithms
 - Analysis of algorithms
- In addition, we will continue to explore good programming practices
 - Good programming style
 - Good habits, such as creating test cases
 - Use of tools, such as Eclipse and Junit
- It will also be necessary to cover more Java

Topics

- Introduction
- Programming Revision
- Methods and Objects
- Arrays and Array Algorithms
- Big O Notation
- Sorting Algorithms
- Stacks and Queues
- Linked Lists
- Recursion
- Bit Manipulation

Learning outcomes

- Recognise the importance of program complexity
- Describe a variety of structures for storing data such as arrays, linked lists, stacks and queues
- Explain a range of algorithms involving searching and sorting
- Identify data structuring strategies appropriate to a given context
- Design, develop, test and debug object-oriented programs in Java
- Apply data structuring techniques to the design of computer programs

Java in this course

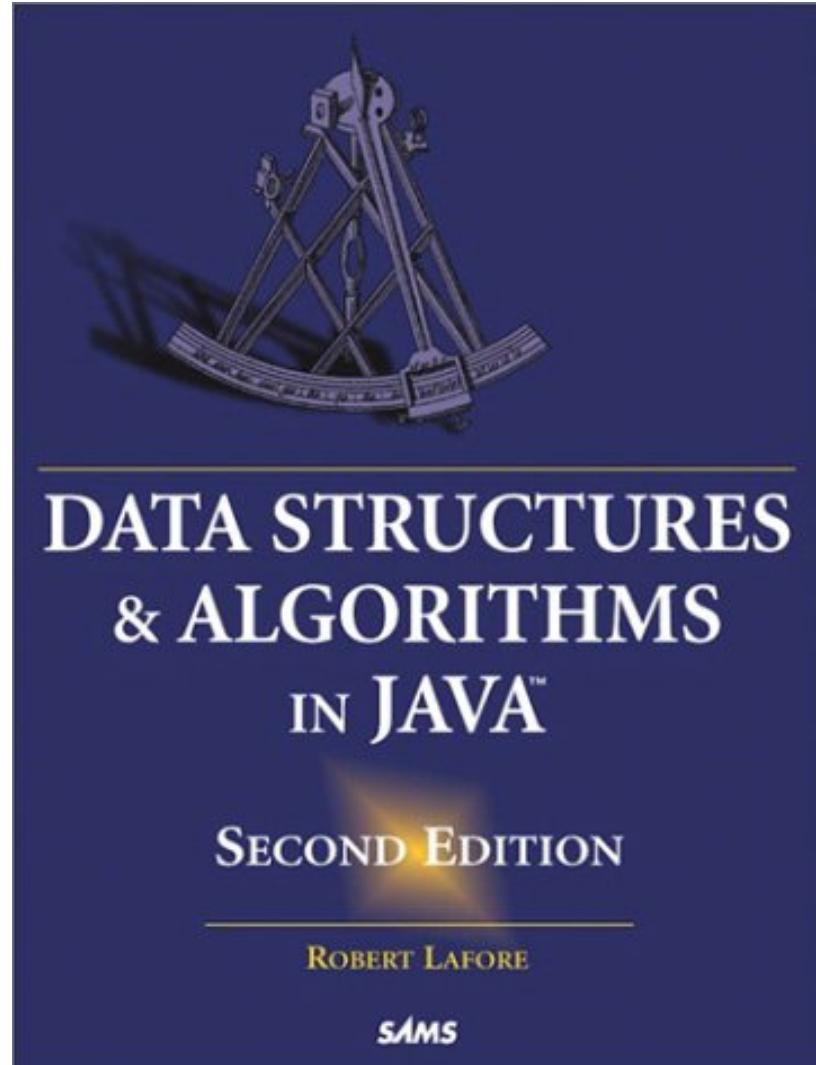
- This is a course in algorithms and data structures, not a second course in Java
- But...
 - Java will be the primary programming language
 - We will study **Java Collections** in detail, as they are extremely relevant to the course
 - You will be expected to use Visual Studio or Eclipse 4.20.0 or newer
 - Moodle VPL: JDK/JRE 7 - java version "1.7.0"

Java Collections

- Java Collections implement many of the most important data structures for you
 - A traditional data structures course would have you implement these yourself
 - I don't believe in re-inventing the wheel
 - However, you need to know how these data structures (or built-in functions) are implemented, for the times when you need something more than Java gives you
 - e.g., [Arrays.sort\(int\[\] arr\)](#)

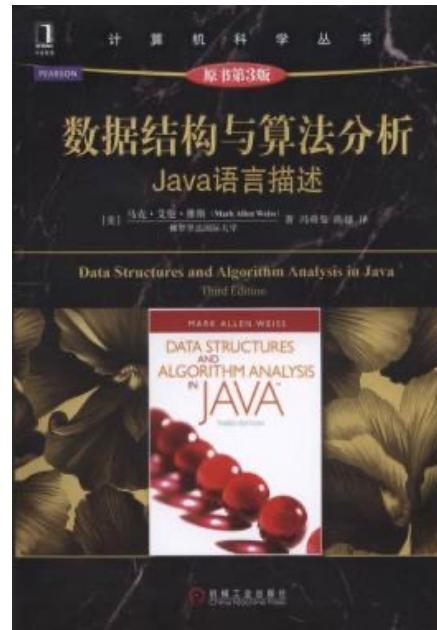
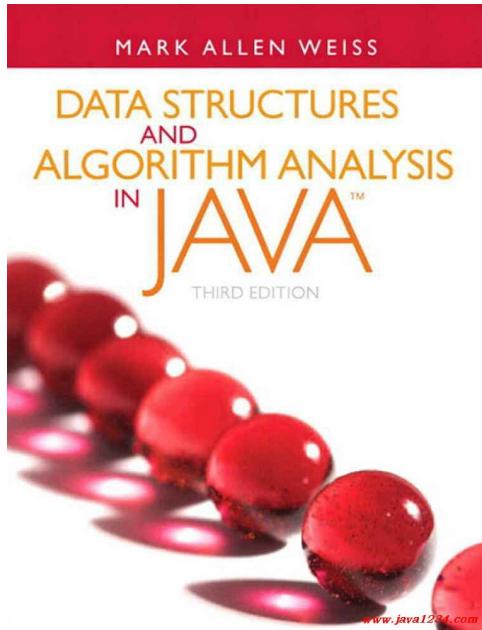
Required textbook

- *Data Structures & Algorithms in Java*, Second Edition, by Robert Lafore
- This book has the clearest and most understandable explanations of algorithms that I have ever seen
- It does not, however, cover some of the essential math we will need



Recommended book

- Data Structures and Algorithm Analysis in Java, 3rd Edition, Mark Allen Weiss
- 数据结构与算法分析：Java语言描述（原书第3版），马克·艾伦·维斯 著，陈越 译，出版社：机械工业出版社
- Reasons for recommendation: it has Chinese version



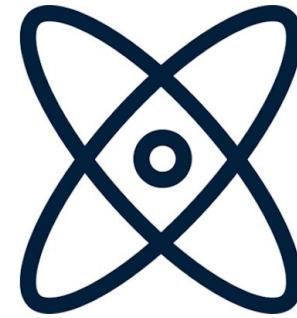
Learning with practice



<https://leetcode.com/>



<https://www.geeksforgeeks.org/>



<https://www.algoexpert.io/>

Outline



- Classes & Labs
- Course content
- **Grading**
- Tools to be used in this course
- Motivation

Grading

- We will have:
 - Several quizzes (10%)
 - Several programming assignments (30%)
 - No midterm
 - One final exam (60%)

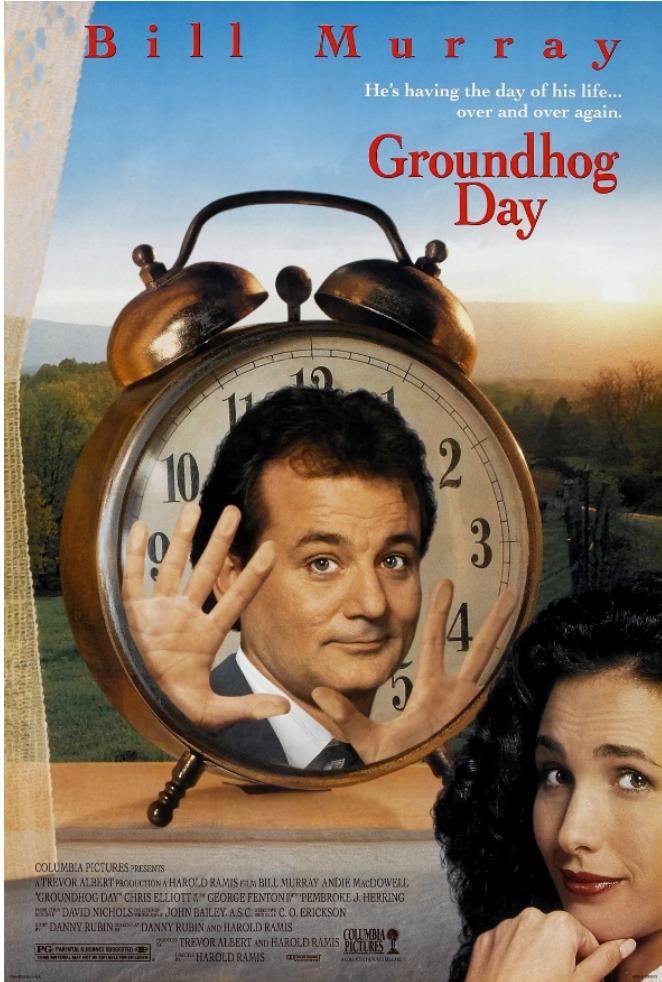
Assignments (and quizzes)

- Will be released on Moodle
- Except as otherwise noted, all assignments:
 - Are to be done by yourself.
 - As before,
 - You may discuss the assignments with other students
 - You may help (and get help with) debugging
 - You may **not** give your source code to anyone

Assignments

- I think that each assignment can be done on our lab (approx. 3hours, 8 labs).
- Late assignments will receive a deduction of 20% of the marks, and may not be accepted if more than a week late

Attendance



- High exam failure is directly correlated with low attendance.
- Do not drop anchor here!!!
- Same goes for Repeat students
- Take responsibility for your own Learning!

Plagiarism – Zero Tolerance

[Avoiding Plagiarism- Library Video Guide](#)

[Plagiarism Policy](#)

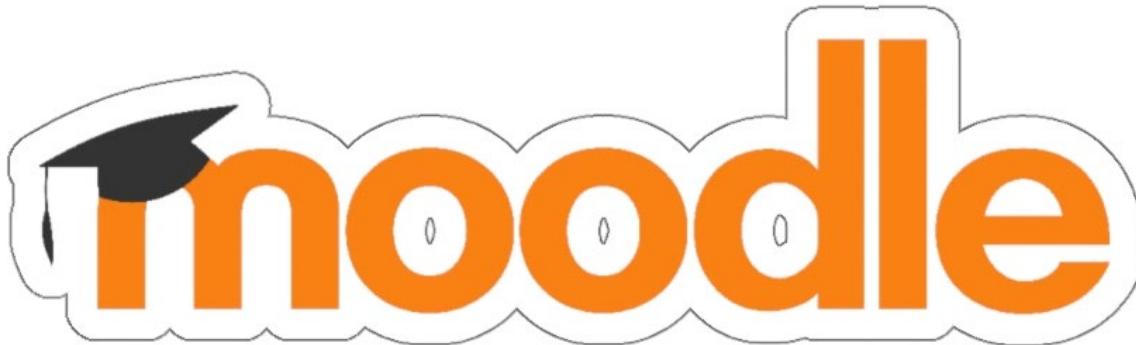


Outline



- Classes & Labs
- Course content
- Grading
- **Tools to be used in this course**
- Motivation

We will use



TurningPoint

Virtual
Programming Lab

- Announcements
- Slides
- Quizzes
- Assignments



<https://tppoll.eu/p/102246>

Check your Moodle forum digest email

Moodle 2021: forum digest

D Do not reply to this email (via Moodle 2021) <no-reply@moodle.mu.ie> [↶](#) [⤵](#) [⤶](#) [⤷](#) ...

Thu 20/05/2021 20:18

To: Guangyuan Piao

This is your daily digest of new posts from the Maynooth University Moodle 2021 forums. To change your default forum email preferences, go to [your user profile](#).

[CS183FZ\[A\].\(20-21:S2\)](#) -> [Forums](#) -> [Lecturer Announcements](#) -> [Weekly period from next week](#)

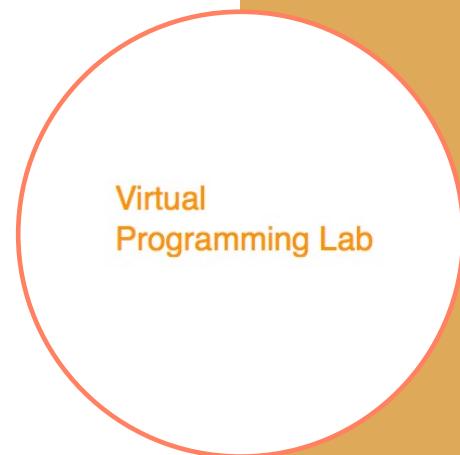
[Weekly period from next week](#) by [Guangyuan Piao](#) - Thursday, 20 May 2021, 2:17 PM

Everyone is subscribed to this forum [Change your forum digest preferences](#)

[EE180FZ\[A\].\(20-21:S2\)](#) -> [Forums](#) -> [Lecturer Announcements](#) -> [Weekly period from next week](#)

Virtual Programming Lab (VPL)

- Edit online
- Run your code
- Evaluation & Auto grading
- Submission view
- Test case setup
- Submit your file from disk



Outline



- Classes & Labs
- Course content
- Grading
- Tools to be used in this course
- **Motivation**

Meme



Meme

@bejewelledbud

Can you guys please recommend books that made you cry?

@FreaseDaddy

Data Structures and Algorithms in Java (2nd Edition) 2nd Edition

by Robert Lafore ▾ (Author)

★★★★★ 114 customer reviews

Look inside

Kindle \$29.80 Hardcover \$33.89 - \$45.04 Paperback \$23.39 - \$27.18 Other See all 6 versi

Buy used

Meme



Who is putting all the Algorithm
books in the horror section?"

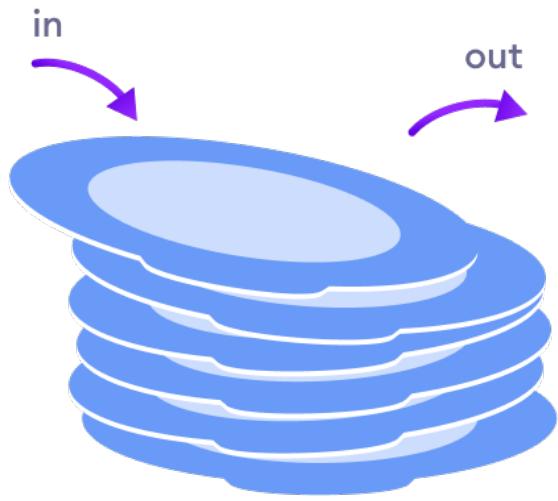


We will
keep it
simple

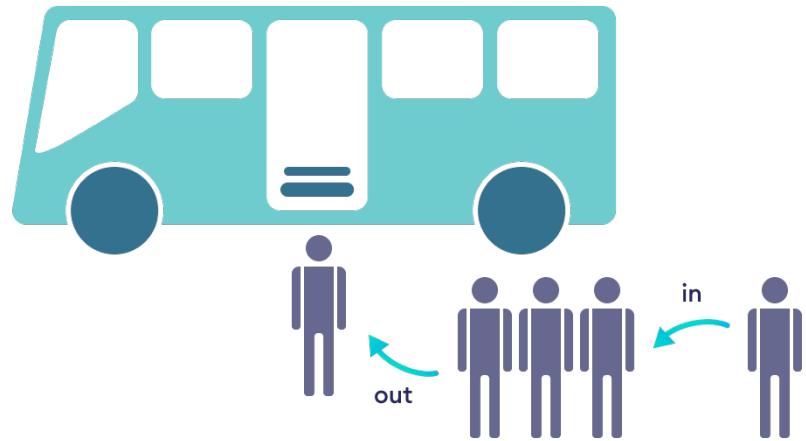
What is...

- **Algorithm:** A step-by-step procedure for solving a problem via a computational process
- **Program:** An implementation of an algorithm in some programming languages
- **Data Structure:** A conceptual system for organizing the data needed to solve some problems

Real life examples



A pile of plates



Real life examples

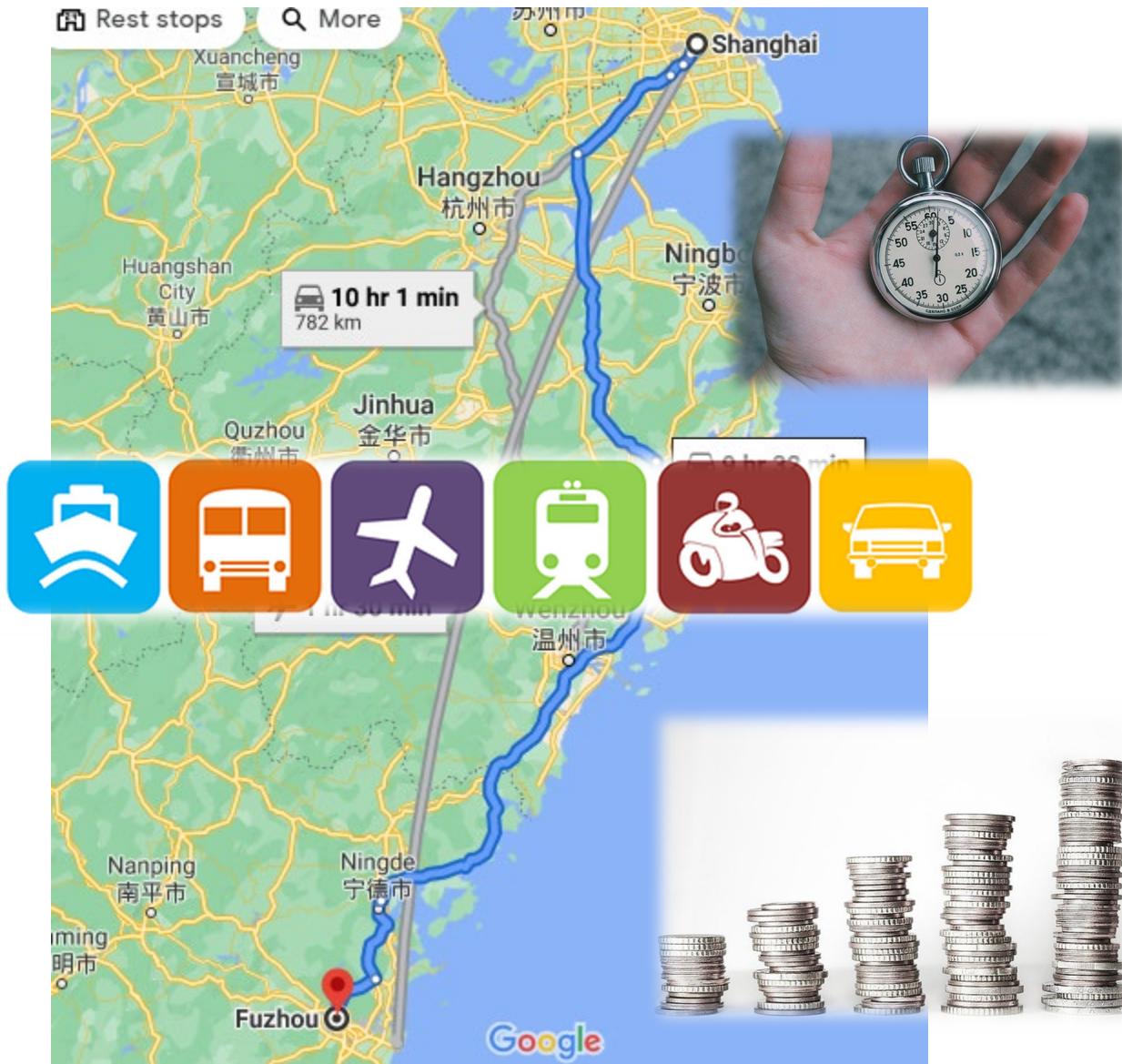


Source: <https://www.workfront.com/blog/work-organization-tips-30-to-take-you-from-mess-to-master>



Source: <https://etiquettejulie.com/organize-your-office-productivity/>

Real life examples





Real life examples

- Most traffic lights don't automatically cycle through green, yellow, and red. Rather, there are sensory inputs that determine the signals' timing based on the flow of traffic.
- The algorithm is a well-constructed, step-by-step order that directs the traffic appropriately (although it may not feel like it when you're sitting at a red light).

Real life examples

- .png, .mp3 files (data compression algorithms)
- Assigning the nearest Uber driver
- Sorting contacts in your app
- Plagiarism detection
- ...
- Linux file system
- Scheduling processes in OS
- Customer care call management
- ...

Why algorithms and data structures?

- Apply the right set of tools to solve the given problem efficiently

Why algorithms and data structures in the context of industry/academic settings?

Why algorithms and data structures?

- Grade

Why algorithms and data structures?

- Grade
- Further graduate studies/research

COURSES:

PREREQUISITE

Analysis of **Algorithms** is the core of Computer Science, which unites the many disparate sub-fields. All doctoral students are expected to have completed an acceptable graduate or upper-level undergraduate lecture course in Analysis of **Algorithms**, with grade B+ or higher, prior to entering the program.

Source: <https://www.cs.columbia.edu/education/phd/requirements/>

Why algorithms and data structures?

- Grade
- Further graduate studies/research

4.5.1 The Breadth Requirement

Students must demonstrate sufficient knowledge of some diverse areas of computer science.

Students must take two courses from each of the following two categories:

A	B
<ul style="list-style-type: none">1. 509 (Introduction To The Foundations Of Computer Science)2. 510 (Numerical Analysis)3. 513 (Design And Analysis Of Data Structures And Algorithms I)4. 514 (Design And Analysis Of Data Structures And Algorithms II)5. 521 (Linear Programming)6. 522 (Network And Combinatorial Optimization Algorithms)7. 529 (Computational Geometry)	<ul style="list-style-type: none">1. 505 (Computer Structures)2. 507 (Advanced Computer Architecture)3. 515 (Programming Languages And Compilers I)4. 516 (Programming Languages And Compilers II)5. 519 (Operating System Theory)6. 520 (Introduction To Artificial Intelligence)7. 523 (Computer Graphics)8. 525 (Brain-Inspired Computing)9. 530 (Principles of AI)10. 532 (Logical Foundations Of Knowledge Representation)11. 533 (Natural Language Processing)12. 534 (Computer Vision)13. 535 (Pattern Recognition: Theory And Applications)14. 536 (Machine Learning)15. 541 (Advanced Data Management)

Why algorithms and data structures?

- Grade
- Further graduate studies/research
- Industry settings

Software Development Engineer
Amazon - Dublin, County Dublin, Ireland

Apply Save ...

- Proficient written and oral communication skills.
- Equivalent experience to a Bachelor's degree based on 3 years of work experience for every 1 year of education.
- 5+ years professional experience in software development.
- Computer Science fundamentals in object-oriented design.
- Computer Science fundamentals in data structures.
- Computer Science fundamentals in algorithm design, problem solving, and complexity analysis.
- Proficiency in, at least, one modern programming language such as C, C++, Java, or Perl.
- Excellent problem solving skills.
- Experience building large scale web services.

Why algorithms and data structures?

- Grade
- Further graduate studies/research
- Industry settings

Software Engineer
HSBC - Guangzhou, Guangdong, China

Apply  Save ...

technical design, test plan etc.;

- Work closely with various application teams and local support teams to provide application and technical support;
- Conduct environment setup, system tests, and user acceptance tests;
- Provide system implementation and post-release supports;
- Work closely with HSBC business and IT parties to meet business objectives;

Requirements

- Bachelor or Master degree in Computer Science, IT engineering, Mathematics / Computation or related discipline;
- Strong IT problem solving skills and communication skills;
- Familiar with algorithm. Eager to learn new technologies.
- Good command of spoken and written English.
- Experience on iSeries, SQL, Python or Linux shell scripts development is a plus;
- Knowledge on banking products will be an advantage;
- Fresh graduates are acceptable if qualified.

Why algorithms and data structures?

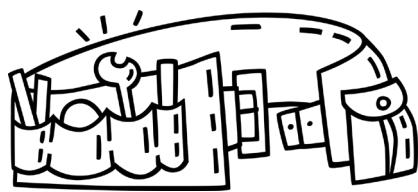
- Grade
- Further graduate studies/research
- Industry settings

Why are data structures so important in technical interviews?



Paul K. Young, Software Engineer at Google

Answered May 23, 2016

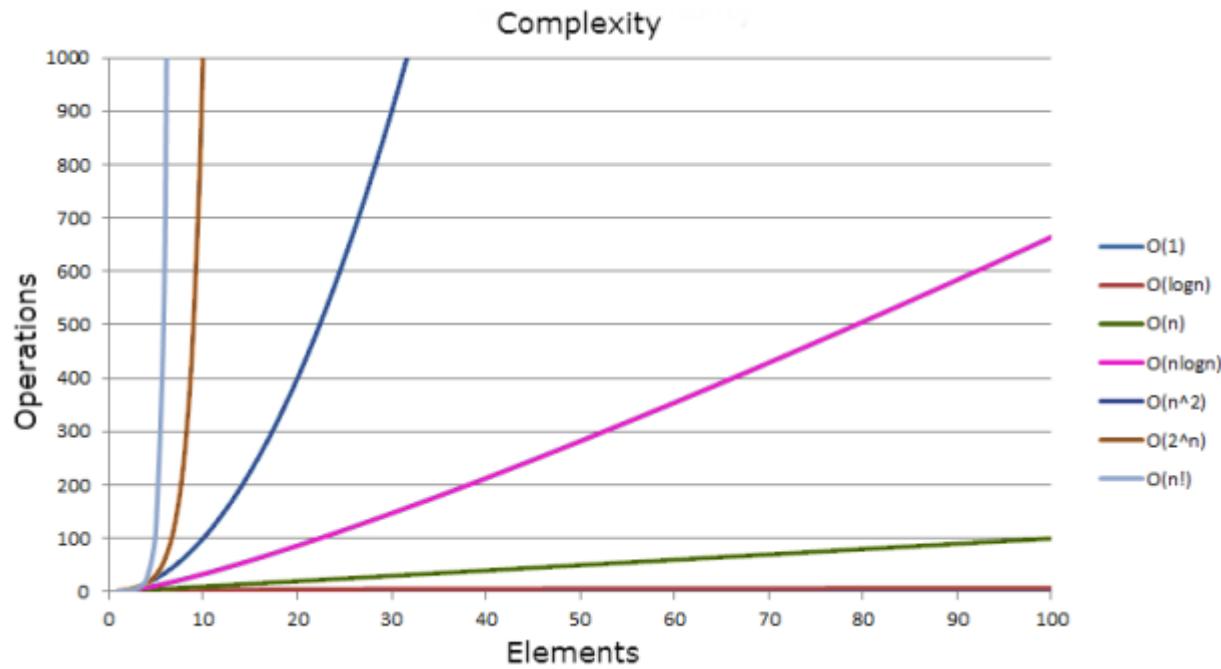


Consider the toolbelt analogy. Having the right tools for the job can make the job a lot easier. Also, being trained in the proper use of the tools is essential to be effective. Knowing which data structure to use for a particular problem and how to correctly apply it can improve performance, resource utilization and minimize the risk of critical failures on edge cases.

Example 1: You want to fit a large dictionary in memory but don't have the space to store all the keys individually. If you know about tries you might be able to squeeze the dictionary in memory. If you don't know about them you might engineer a solution that unnecessarily requires sharding (substantially increasing the complexity) or disk swapping (slowing things down to a crawl).

Why algorithms and data structures?

- **Example:** Suppose you are working in Facebook company. You come up with an optimal solution of a problem (like sorting a list of users from India) with a time complexity of $O(n\log n)$ instead of $O(n^2)$ and assume that n for the problem here for the company in real life scenario is 100 million (very fair assumption considering the number of users registered on Facebook exceeds 1 billion).
- $n\log n$ would be 800 million, while n^2 would be 10^7 billion. In cost terms, you can see that the efficiency has been improved more than 10^7 times, which could be a huge saving in terms of server cost and time.

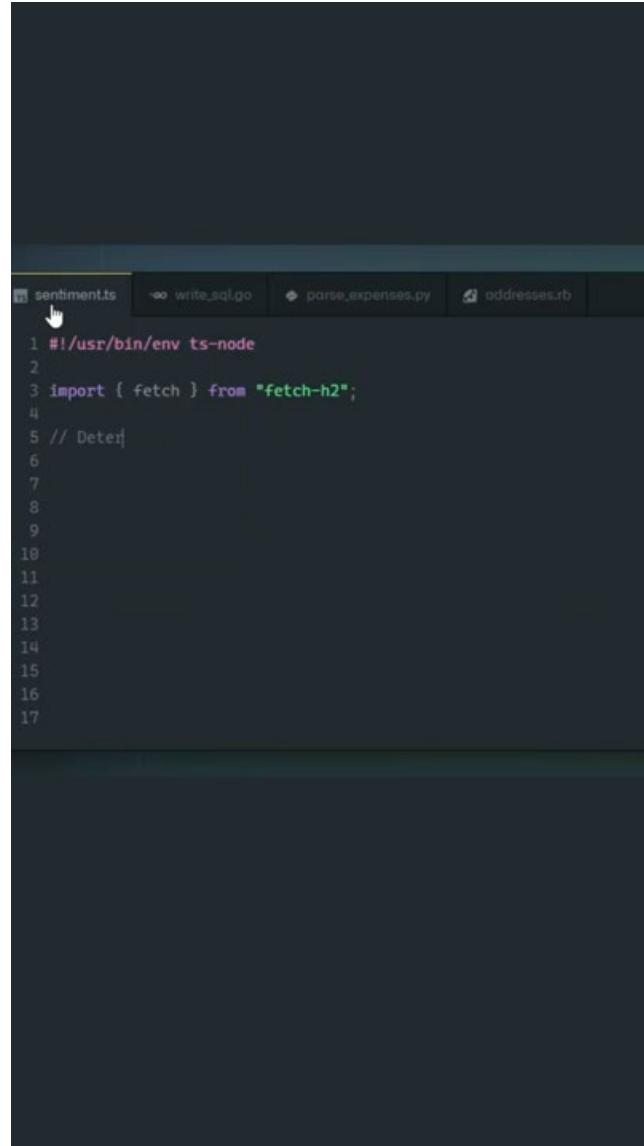


<https://dzone.com/articles/learning-big-o-notation-with-on-complexity>

Why algorithms and data structures?

- Engineers working in Google, Microsoft, Facebook, Amazon-like such companies are different than others and paid higher as compared to other companies...but why?
 - coding/implementation: approx. 20-30% of the time allotted to a project
 - Most of the time goes into designing things with the best and optimum algorithms to save on the company's resources (servers, computation power, etc).

Why algorithms and data structures?



A screenshot of a code editor showing a file named `sentiment.ts`. The code is a Node.js script:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Deter]
6
7
8
9
10
11
12
13
14
15
16
17
```

<https://copilot.github.com/>

Why algorithms and data structures?

- Apply the right set of tools
(algorithms and data structure) to solve the given problem efficiently



TurningPoint

<https://tppoll.eu/p/469528>

