CMPT 225: Data Structures & Programming – Unit 01 – Introduction

Dr. Jack Thomas
Simon Fraser University
Spring 2021

Welcome to CMPT 225!

- A class on data structures, algorithms, programming, analyzing efficiency, and approaching problems like a computer scientist.
- Today's Goals:
  - Discussing how the course will run.
  - An overview of the topics we'll cover.
  - A brief introduction to Java.

The Teaching Team

- Your instructor, Dr. Jack Thomas
  - jackt@sfu.ca
- Your TAs, Xinwei Gu and Carmen Zhuang
  - xinwei_gu@sfu.ca and carmen_zhuang@sfu.ca

Schedule

- **Virtual lectures** Monday, Wednesday, and Friday, from 11:30am to 12:20pm, here on Canvas.
  - Recordings available also on Canvas.
- **Weekly labs** on Wednesday according to your lab section's time.
- **Office hours** on Discord, Carmen on Fridays from 12:30pm to 1:30pm, Xinwei on Mondays from 2 to 3pm.
- As for my office hour, let's decide that now!

The Course Website(s)

- The **main course website**: https://opencoursehub.cs.sfu.ca/jackt/grav-cms/cmpt225-1/home
- **CourSys**, for submitting assignments and labs: https://coursys.sfu.ca/
- **Canvas**, for the lectures, midterm, and exam: https://canvas.sfu.ca/courses/60286
- **Piazza**, for helping each other out: https://piazza.com/class/kjj827ffg0n69u
- **Discord**, for office hours, labs, and meetings: https://discord.com/invite/hCaqhYzjfq

Course Grade Breakdown

- **Assignments** (50%) Five bi-weekly assignments.
- **Weekly Labs** (10%) Ten labs, 1% each, marked for effort.
- **Midterm** (15%) March 8th on Canvas.
- **Exam** (25%) Date TBD, also on Canvas.

Textbook

- Recommended, not Required.
- *Data Structures & Algorithms in Java*, Sixth Edition, by Goodrich, Tamassia, and Goldwasser.

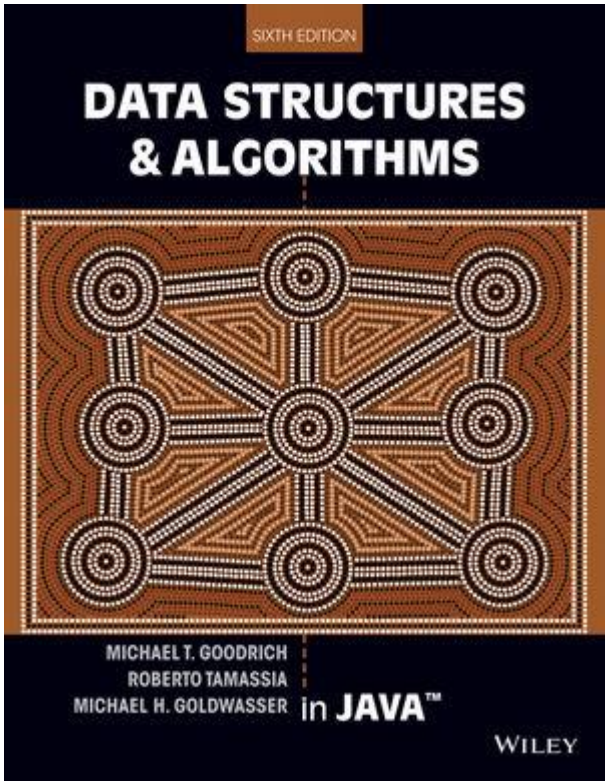
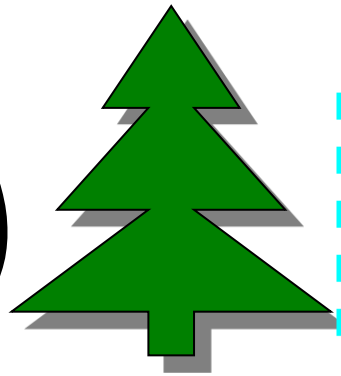


Image credit: https://www.wiley.com/en-ca/Data+Structures+and+Algorithms+in+Java%2C+6th+Edition-p-9781118771334

Introduction to the Introduction: What's a Data Structure, Anyway?

- "A way in which data is stored for efficient search and retrieval" (Encyclopaedia Britannica)
- Essentially, different ways you can organize your code and the data stored in it depending on what you're trying to accomplish.
- Some we'll cover:
  - Stacks
  - Trees
  - Hash Maps



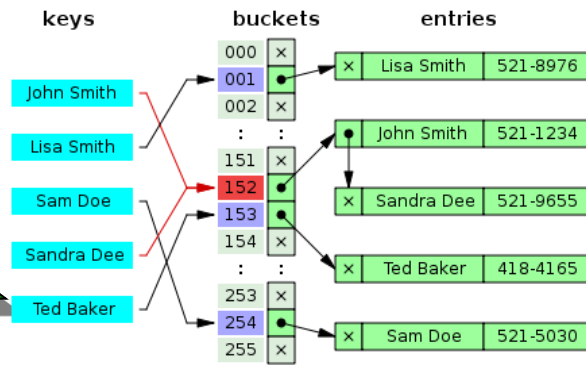


Image credit: https://en.wikipedia.org/wiki/File:Hash_table_5_0_1_1_1_1_1_LL.svg

Example Data Structure: An Array

- A very basic structure built into a lot of programming languages for storing a set of objects of the same type.

```
int[] numbers = {3, 5, 0};
System.out.println(numbers[0]);
```

- The “Array” data structure is more than any one implementation in any one programming language, it’s the whole abstract concept.

How About An Algorithm?

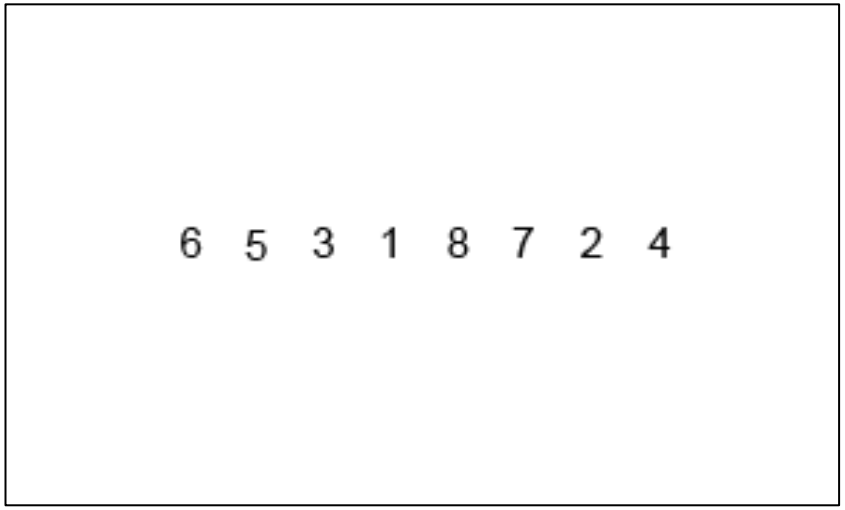
Image credit: https://github.com/sugataach/al-gore-rhythms

No really though

- “A specific procedure for solving a well-defined computational problem.” (Encyclopaedia Britannica)
- There’s a well-studied set of algorithms that apply to a lot of common programming problems, especially around storing and retrieving data.
- This term, we’ll:
  - Introduce different algorithms (e.g. sorting)
  - Learn how to analyze them (e.g. Big-O notation)
  - Discuss when to apply them (needed a third thing to put here for the rule of threes)

Example Algorithm: Merge Sort

- A method for sorting a set of values through splitting them up and sorting as you recombine.



- Can be implemented in different languages to solve different problems, but has general features that are always true.

Image credit: https://en.wikipedia.org/wiki/Merge_sort#/media/File:Merge-sort-example-300px.gif

Topics to be Covered

1. Java
2. Object Oriented Programming
3. Arrays
4. Lists
5. Recursion
6. Analytics
7. Stacks
8. Queues
9. Lists again (surprise!)
10. Iterators
11. Trees
12. Heaps
13. Priority Queues
14. Hash Tables
15. Maps
16. Skip Lists
17. Dictionaries
18. Search Trees
19. Sorting
20. Divide & Conquer
21. Sets
22. Union-Find
23. Selection
24. Dynamic Programming
25. String & Pattern Matching

The Real Goal: Problem-Solving Like a Computer Scientist

- You **don't** want to **reinvent the wheel** every time you need to write a new program.
- Most problems have **well-understood solutions**, which combine the right **data structure** and **algorithm** to produce an **optimal** result.
- What makes a **good** programmer is recognizing **when** to apply an algorithm and data structure. A **great** one will understand **why**.

- **All coding** for this course should be in Java.
- An **object-oriented** programming language.
- Very similar to **C++**, if you're familiar.
- Either way, don't worry, the course website has some **tutorials** to help you get started!
- However, **this is not a course about learning Java** – it's just what we'll use to study the concepts being introduced.

Image credit: https://www.itprotoday.com/programming-languages/java-14-improves-runtime-visibility-overall-performance

How to Get Started with Java

- The course website includes some basic tutorials, including a chapter on going from C++ to Java.
- **Recommended**: IntelliJ IDEA Community Edition https://www.jetbrains.com/idea/download/#section=windows
- Confirm expected project settings like build system and SDK at the start of each assignment, default to **SDK Java 14** and **no build system**.
- Check out the style guide as well!

Finally, Some Acknowledgements

- Thank you to Dr. John Edgar and Dr. Tom Shermer, SFU Lecturers who have previously taught this course and whose material helped inform and develop my own version of the course.
- Thank you to the authors of the aforementioned *Data Structures & Algorithms* book, which this course heavily draws on.
- And thanks to you, for joining me today!

Recap – The End of the Beginning

- This course will introduce you to the **data structures** and **algorithms** you'll need to **efficiently solve** most programming problems.
- The course expectations have been laid out (complete the five **assignments**, attend the ten **labs**, study hard for the **midterm** and **exam**!).
- Now go to the course website, join the Discord, download the IDE, watch the Java tutorials, hit subscribe, leave a review, give five stars.
- **Starting Wednesday**: Object-Oriented Design.
- No lab or assignment this week!