

# CSC 172– Data Structures and Algorithms

Lecture #1  
Logistics and Introduction  
Spring 2018  
TuTh 3:25 pm – 4:40 pm

# Agenda

- Administrative aspects
- Brief overview of the course

Chapter 1

# **ADMINISTRATIVE ASPECTS**

# Teaching Staff

- Instructor:
  - Tamal Biswas
  - [tbiswas2@cs.rochester.edu](mailto:tbiswas2@cs.rochester.edu)
- Graduate TA:
  - Divya Ojha
  - [dojha@cs.rochester.edu](mailto:dojha@cs.rochester.edu)
- Undergraduate TAs and Workshop Leaders:
  - You will meet them in labs and workshops ☺

# Q & A

- Q: Will there be any workshop next week?
- A: The first workshop will be on January 28<sup>th</sup> (Sunday)
- Q: Will there be any lab next week?
- A: Yes. Starting from this coming Monday, January 22<sup>nd</sup>

# Q & A

- Can I go to any Lab or Workshop?
- It depends.
- We discourage switching workshops/labs as that makes accountability harder.
- But, if you must do,
  - please go to the corresponding workshop/lab and let the workshop leader/Lab TAs know your intent.
  - Only if they permit, you may do the switch.

# When/where/how to talk to me

## Algorithm 1:

You can ask questions during the class

else

Piazza (Most effective)

[www.piazza.com/rochester/spring2018/csci72/home](http://www.piazza.com/rochester/spring2018/csci72/home)

else

Contact TAs

else

Come during office hours

(Tu 12pm - 1pm and We 11am - 12pm)

Location: Wegmans 2107

else

Email me. (The last resort and most ineffective)

Email: [tbiswas2@cs.rochester.edu](mailto:tbiswas2@cs.rochester.edu)

Subject line should start with CSC172:

# Prerequisites

- CSC 171
  - Object Oriented design/programming concepts
  - You must have done some programming before!
- MTH 150
  - Mathematical maturity
  - Recursion, trees, graphs



# Course Materials

- Course Website
  - <http://www.cs.rochester.edu/courses/172/spring2018/>
- Textbook:
  - <https://people.cs.vt.edu/shaffer/Book/JAVA3elatest.pdf> (free pdf download)
  - Web version:  
[http://lti.cs.vt.edu/LTI\\_ruby/Books/CS172/html/](http://lti.cs.vt.edu/LTI_ruby/Books/CS172/html/)



# Recordings of the Lectures

- Will be soon available on Blackboard.

# OneNote

- OneNote is our online ‘white board’
- It’s not for review
- Hint: When I write something, it’s probably time to keep a note of that too.
- Link: accessible from course website

# Work Load

- Heavy!
- Approximately 30 pages of reading per week
- 1 Midterm
- 1 Final
- +/- 3 projects
- A quiz every week

# Grading Policy

- **Quiz (10%)**
  - Not more than 12 Quizzes (each worth 2%)
  - Top 10 quizzes will be considered
- **Exam component (35%)**
  - 1 midterm exam (15%)
  - 1 final exam (20%)
- **Projects, Labs, and Workshops (55%)**
  - 3+ projects (35 %)
  - Labs: 15%
  - Workshop Participation: 5%
- **Class Participation**
  - Ask Questions in Class. Take part in Discussion
  - Provide Feedback to improve class
  - Exemplary students will get at max 3 bonus pts for class/piazza participation.
- **Late submission:**
  - No Late submission allowed
- **Incompletes & Make-up exams**
  - Not given except in provably extraordinary cases! (see syllabus)

# Letter Grades

Percentage score	Letter grade
90-100	A
85-89	A-
80-84	B+
75-79	B
70-74	B-
65-69	C+
60-64	C
55-59	C-
50-54	D
0-49	E



**teachers  
call it  
cheating,  
we call it  
teamwork**

# Academic Honesty

- On plagiarism:
  - We will report
    - Instructor Resolution Warning Letter
    - Instructor Resolution With Penalty
    - Board Resolution
  - Note: A student who cheats receives a grade lower than a student who did not hand in a project
- We will check you code using Moss
  - (A System for Detecting Software Similarity)
  - <https://theory.stanford.edu/~aiken/moss/>
- Please refer
  - <https://www.rochester.edu/college/honesty/policy.html>
- <https://www.rochester.edu/college/honesty/>



# Academic Honesty

- Group study/discussion is encouraged, but the submission must be your own work!
- Programming:
  - Discussions of ideas are welcomed, but no exchange of codes
  - If you use a piece of code from Mr. Google, say so!

# Q & A

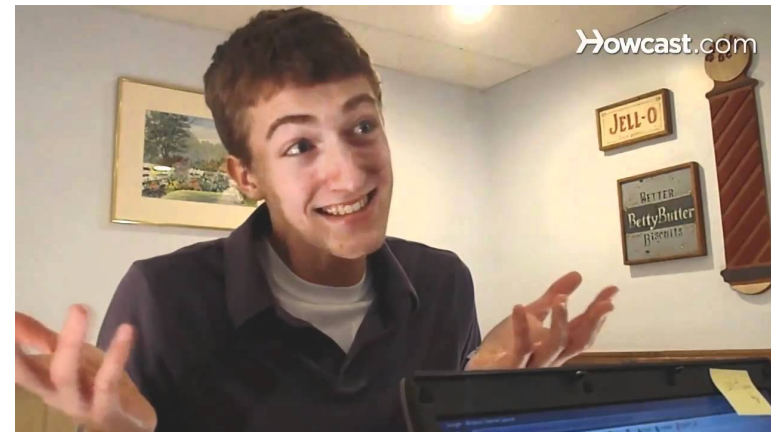
- I found the solution for the project online and copied the whole and provide link to the source. Is it acceptable?
- No. Even though you copy code from elsewhere with references, the code should not be more than 10% of total number of lines

# Q & A

- Then how and when can I use code with references?
- Mostly, a particular function that's hard to implement.

# No Lame Excuses, Please

- I want to go home early, can I take the final early?
- I had a fight with my girlfriend
- I've studied very hard, I understood the stuff very well, but I got a C-, please consider giving an A-



# How to do well in the course?



How to do well in the course?

# study

*(verb)*

The act of texting, eating  
and watching TV with an  
open textbook nearby.

# How to do well in the course?



# How to do well in the course?

- Participate: discuss, answer, ask questions  
“the only stupid question is the question you don’t ask”
- Give suggestions, please! I’ll take them seriously
- Attend all the labs and workshops
- Study every week
- Start early!



Chapter 2

# **BRIEF OVERVIEW OF THE COURSE**

# What you will learn from this course

- Data structures
  - Design
  - Analyze
  - How to use them (with algorithms)
  - When to use them, and which one(s)
  - How to implement (some of) them in Java
- This is *not* a Java course
  - Some Java is covered, sufficient for our above purposes

# What you will not learn from this course

- A lot!
- Why?
  - Both *data structures* and *Java* are huge subjects
  - [https://en.wikipedia.org/wiki/List\\_of\\_data\\_structures](https://en.wikipedia.org/wiki/List_of_data_structures)

# Why Data Structures in Java

- Because you all know Java!
  - CSC 171 is one of the prerequisites
- Data structures are everywhere
  - Numerous examples
- Java is **powerful, fast, widely used**, and (almost) pure object-oriented programming language
  - [http://www.theregister.co.uk/2011/06/03/google\\_paper\\_on\\_cplusplus\\_java\\_scala\\_go/](http://www.theregister.co.uk/2011/06/03/google_paper_on_cplusplus_java_scala_go/)
  - [http://www.fixoncloud.com/Home/LoginValidate/OneProblemComplete\\_Detailed.php?problemid=534](http://www.fixoncloud.com/Home/LoginValidate/OneProblemComplete_Detailed.php?problemid=534)
  - <http://forums.xkcd.com/viewtopic.php?t=108685>
  - <http://www.geeksforgeeks.org/java-not-purely-object-oriented-language/>
- The combination is great for your future **job search!**
  - <http://www.geeksforgeeks.org/top-25-interview-questions/>
- Both are extremely interesting topics!

# Q & A

Q: Do I need to write programs on paper?

A: Yes. In Exams and Quizzes, you may need to write programs on paper.

- The purpose is not to see how good a programmer you are
- The purpose is solely to check your understanding

Right now, Java is the only tool we have to show our understanding.  
Once you take CSC 282, you will learn how to write algorithms independent of any programming language

# CSC 171: Introduction to Computer Science

CSC 171 covered:

- Operators / loops / methods
- arrays / strings
- Exceptions / debugging
- File input / file output
- search algorithms
- Sorting algorithms
- Object-oriented design / polymorphism / interfaces / inheritance

# CSC 171: Introduction to Computer Science

These were the basics of programming

- The ability to manipulate the computer to perform the required tasks

You saw data storage techniques:

- Arrays, and
- Linked lists
- 

You saw array accessing/manipulation techniques:

- Searching, and
- Sorting

# CSC 172: The Science of Data Structures

In this course, we will look at:

- *Algorithms* for solving problems efficiently
- *Data structures* for efficiently storing, accessing, and modifying data

We will see that all data structures have trade-offs

- There is no *ultimate* data structure...
- The choice depends on our requirements



# CSC 172: The Science of Data Structures

Consider accessing the  $k^{\text{th}}$  entry in an array or linked list

- In an array, we can access it using an index `array[k]`
- We must step through the first  $k - 1$  nodes in a linked list

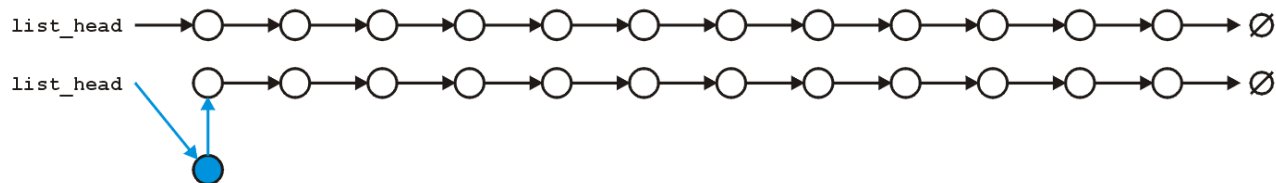
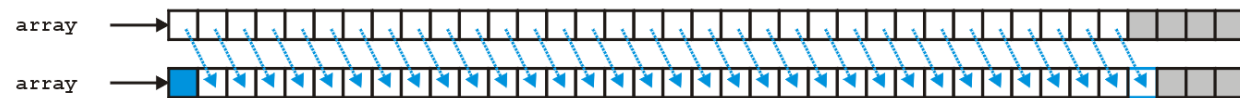
Consider searching for an entry in a sorted array or linked list

- In a sorted array, we use a fast binary search
  - Very fast
- We must step through all entries less than the entry we're looking for
  - Slow

# CSC 172: The Science of Data Structures

However, consider inserting a new entry to the start of an array or a linked list

- An array requires that you copy all the elements in the array over
  - Slow for large arrays



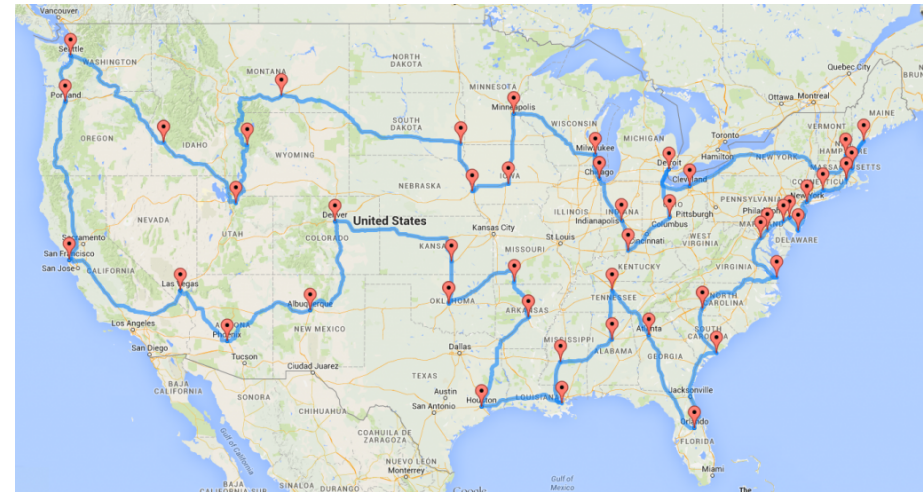
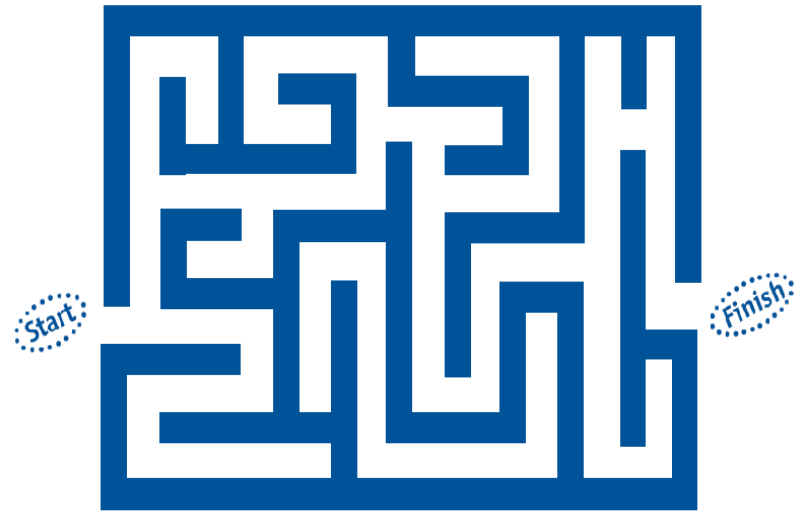
- A linked list allows you to make the insertion very quickly
  - Very fast regardless of size

# CSC 172 Goal

- Find the best data-structure for the problem you are solving
- How can we define best???
- Given the code for an algorithm, you will be able to estimate how much **time** and **memory** it will take to process a given amount of data?

# Side-effects

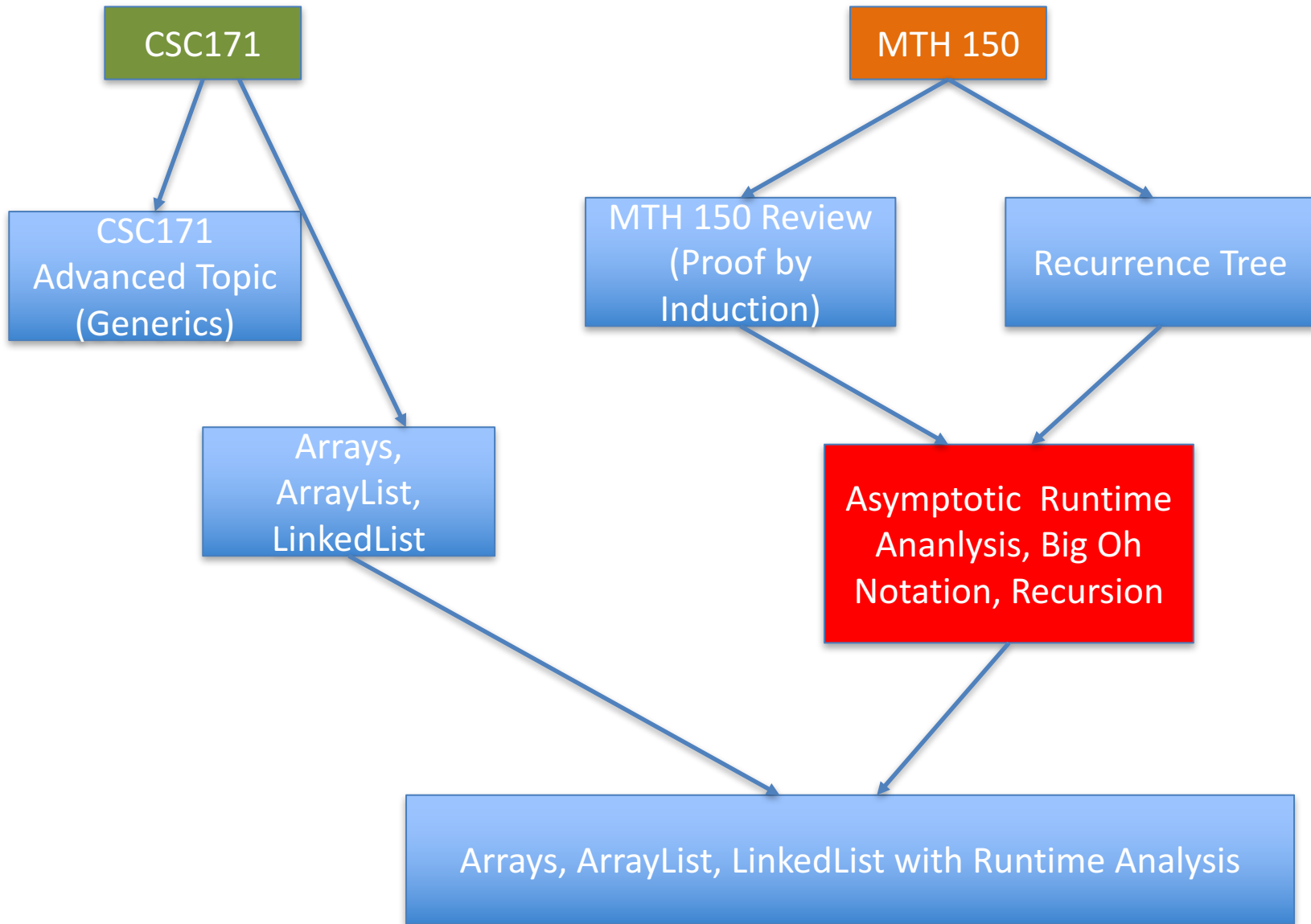
- Become more proficient in coding!
- you should be able to:
  - Write a program to find a path through a maze
  - Find the shortest path
  - And many more interesting problems



Directed Acyclic Graph (DAG) representation

**WHAT WILL WE LEARN?**

# DAG



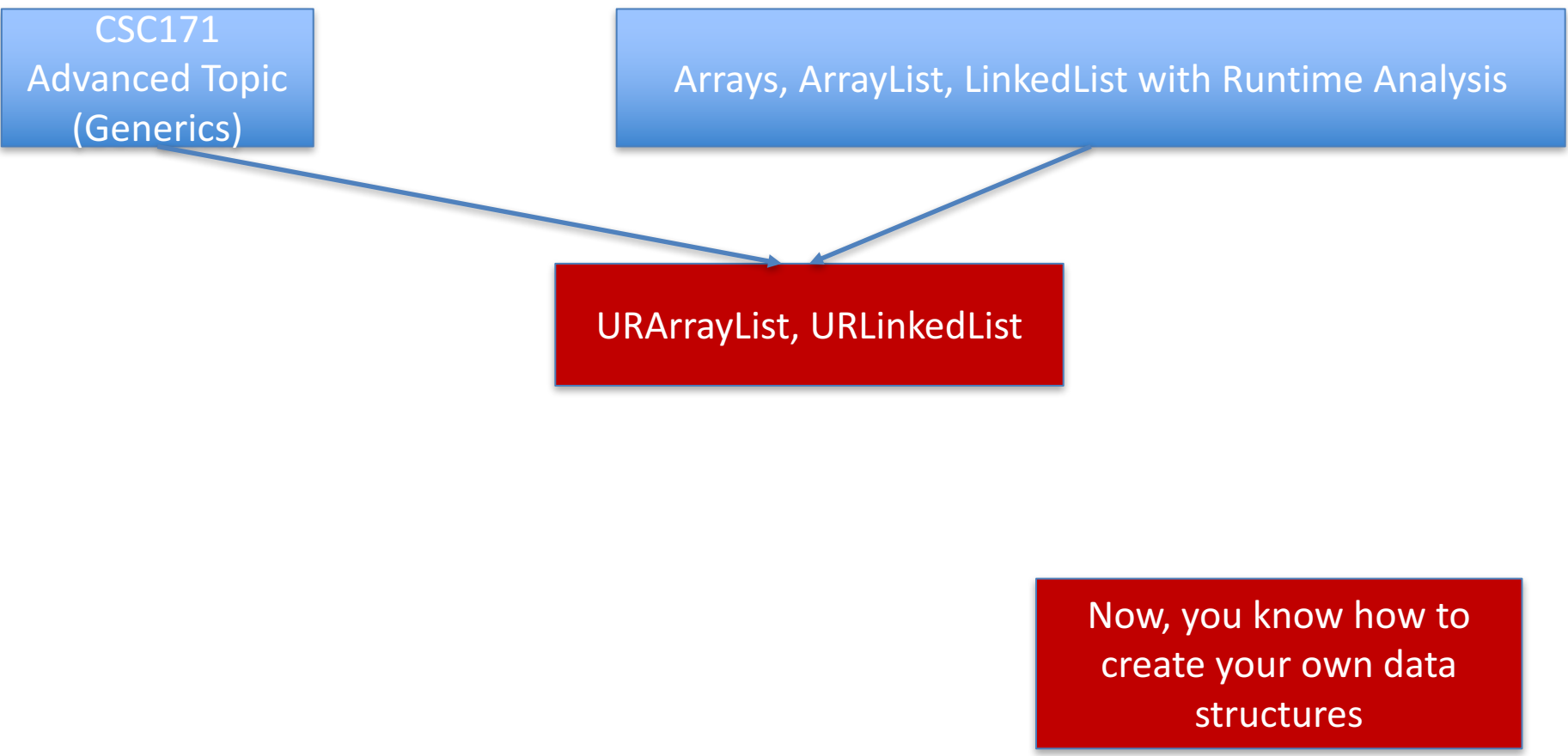
# DAG (Continued)

CSC171  
Advanced Topic  
(Generics)

Arrays, ArrayList, LinkedList with Runtime Analysis

URArrayList, URLinkedList

Now, you know how to  
create your own data  
structures



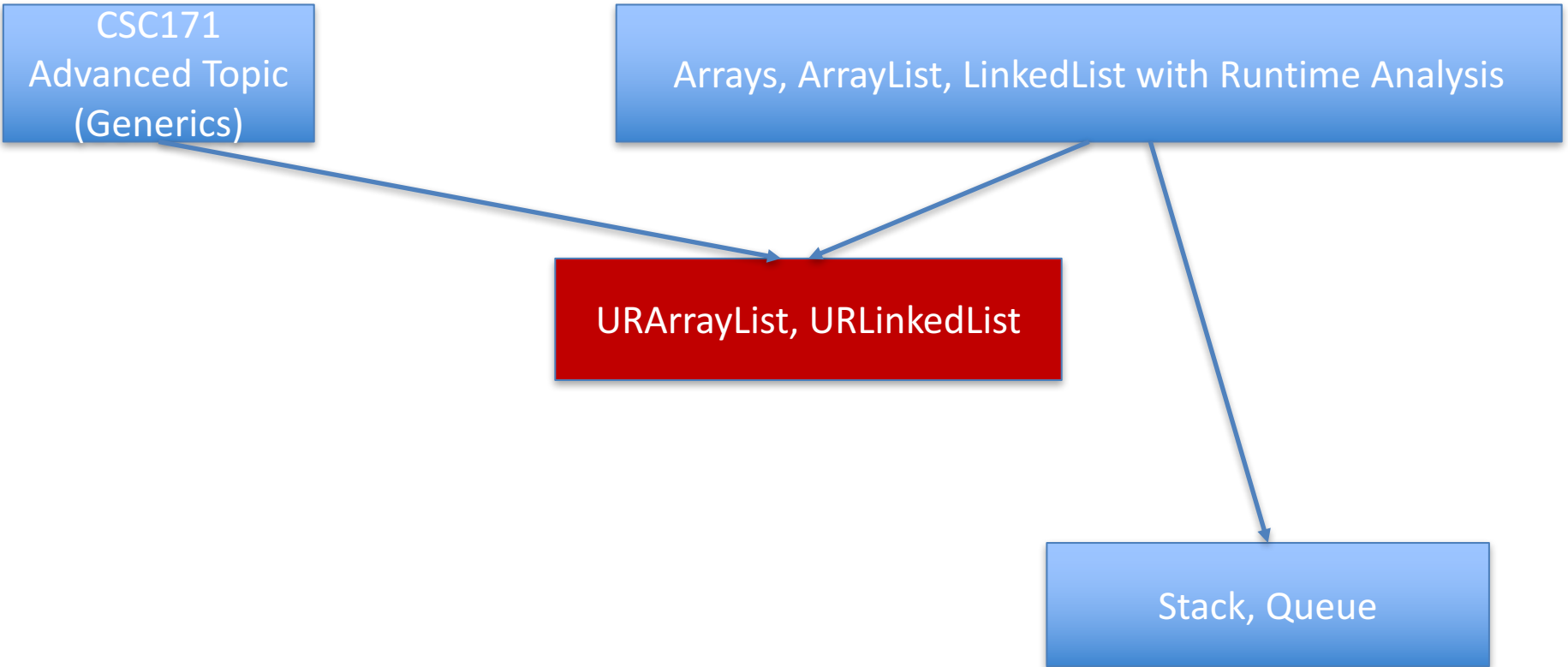
# DAG (Continued)

CSC171  
Advanced Topic  
(Generics)

Arrays, ArrayList, LinkedList with Runtime Analysis

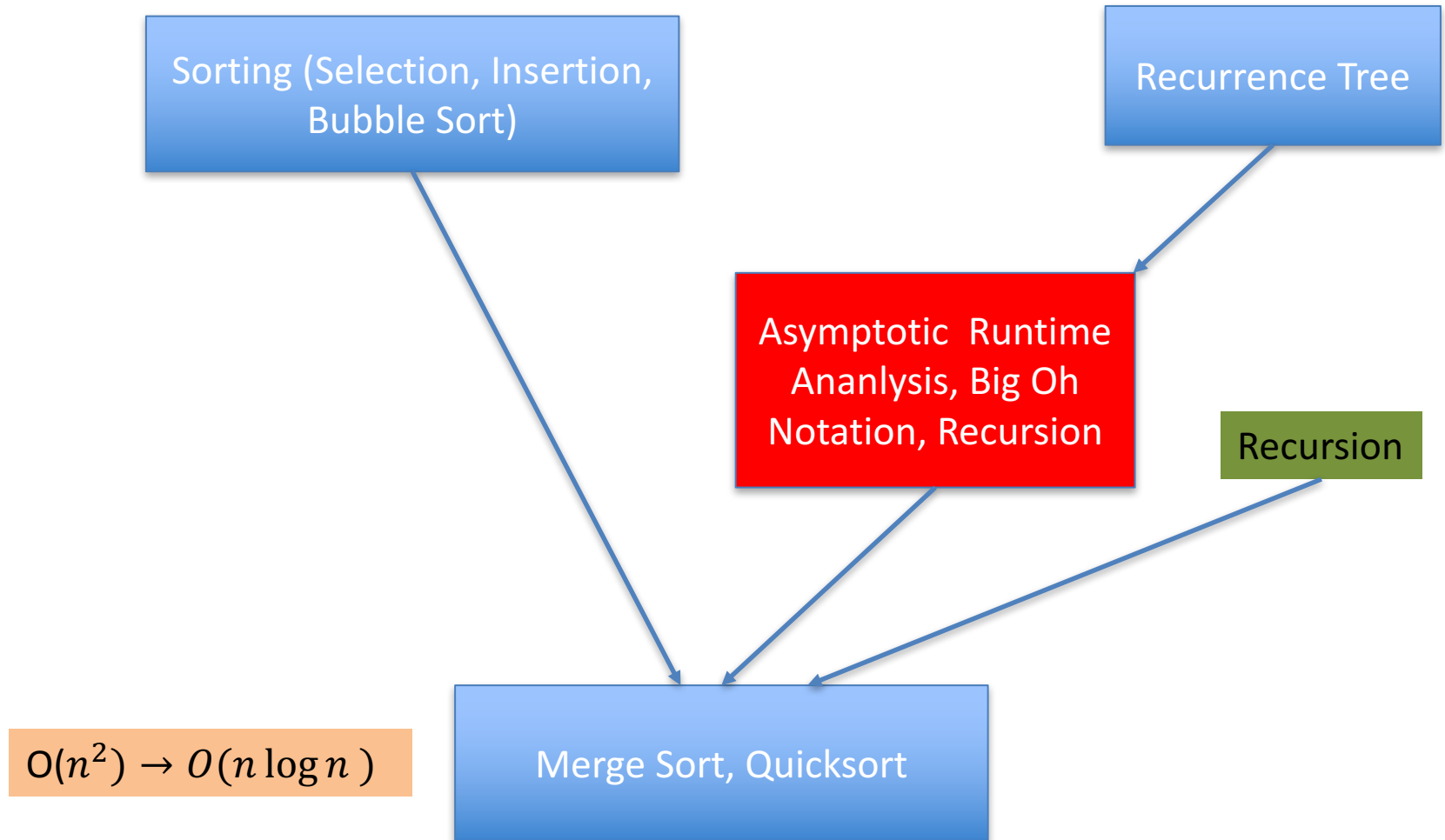
URArrayList, URLinkedList

Stack, Queue

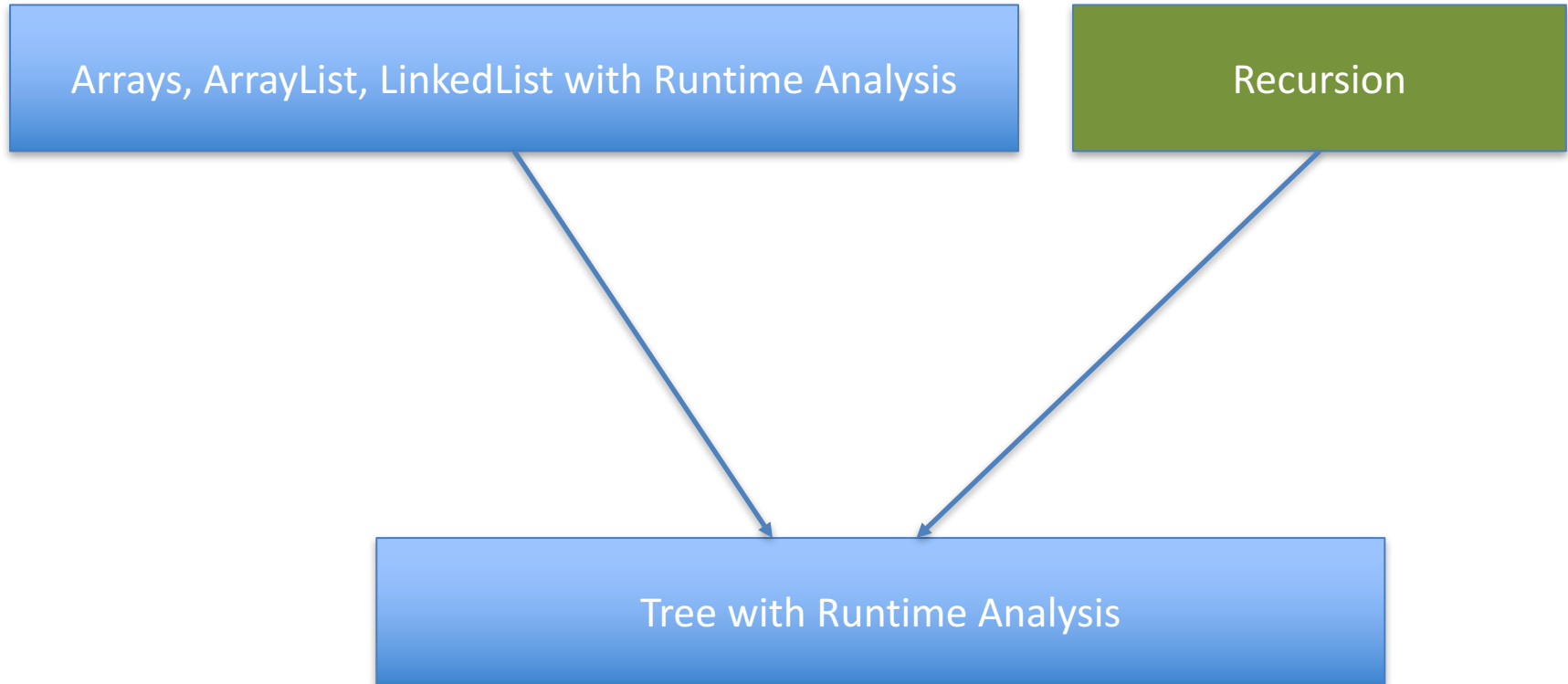




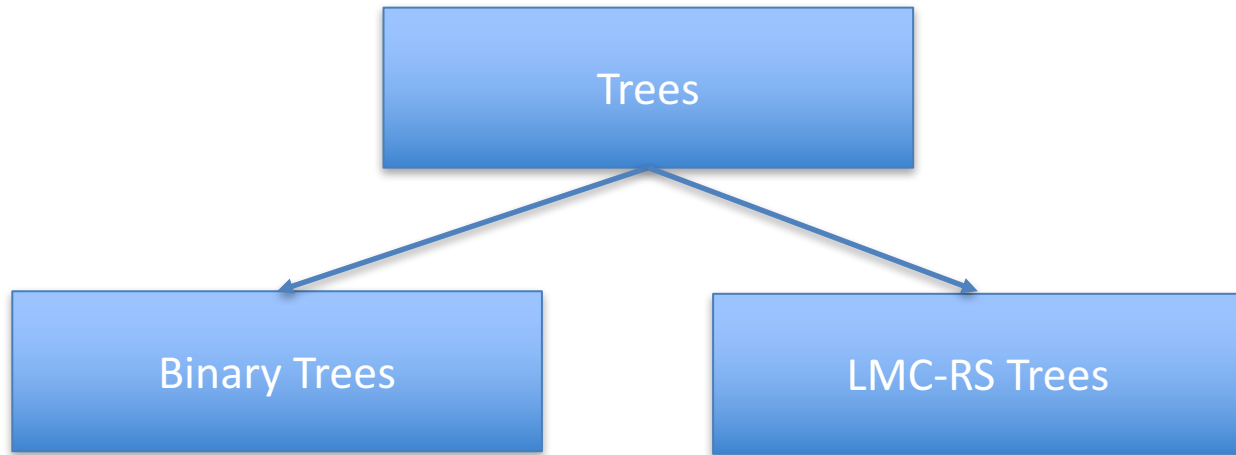
# Sorting



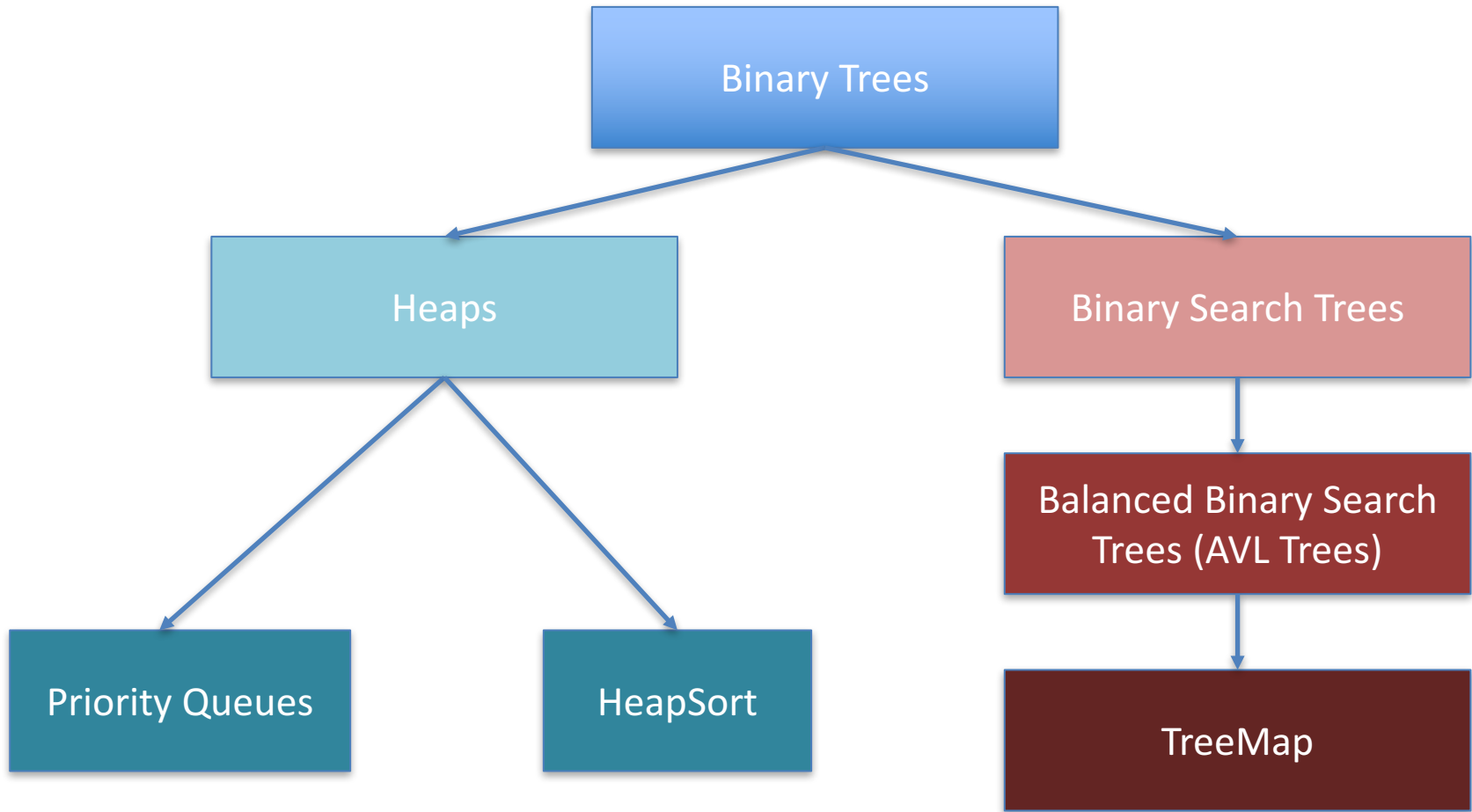
# Linear to Non-linear



# Tree Variations



# Binary Trees



# Last Two Data Structures

Graphs

HashMap

# Outline

- **2 weeks**: the procedural parts of Java, programming environment, some algorithms
- **3 week**: asymptotic notations, sorting, recursion, proof by induction
- **3 weeks**: List, Stack, Queue
- **2 weeks**: Trees
- **1 week**: Hash Tables
- **2 weeks**: Graphs
- **1 week**: Advance topics (if time permits)

# What You Must Do This Week!

- If you have not signed up yet:
  - [www.piazza.com/rochester/spring2018/csci72](http://www.piazza.com/rochester/spring2018/csci72)
- Study
  - Java Generics (Important)
  - <https://docs.oracle.com/javase/tutorial/java/generics/>
  - Chapter 1 and Chapter 2 from eTextBook
    - Chapter 1 (Important)
    - Chapter 2 (Read just for fun)

# Last Words before we begin

- You'll learn as much from me as I will learn from you
  - Give your feedback via email
  - I will also set up a google form soon
- Welcome, again!