



GEORGE MASON
UNIVERSITY®

CS 310 – Fall 2025

Data Structures

L01-Overview

Archange G. Destiné
adestine@gmu.edu

Course Organization

- 1. Instructor / TAs
- 2. The **course** objectives / expectations
- 3. **Syllabus** / Tentative **schedule** / **Administrative**
(grades, projects, participation)

01- Instructors / TAs

Archange G. Destiné

Office Hours:

When? Mondays, 12PM-2PM

Where? ENGR 3241

(17) TAs:

How to get assistance?

Piazza? Office Hours?

About me...



Value: Consistency, Effort, Fairness, Knowledge Sharing

Interest: Explainable AI, Neuro-Symbolic Learning, NLP,...

About you...

Why is the initial survey important?

Some pieces of advice ...

We will use **PollEverywhere** for some activities:

- First activity: Your expectation
- Next activity: Initial Survey

About you...

Using PollEverywhere:

<https://pollev.com/adestine>

Fill this survey before the next meeting:

- Proficiency in Java, Debugging Java Code, Recursive Methods, Generics

02- Course Objectives / Expectations

You will :

- Analyze **correctness** and **efficiency** of algorithms
- Create and **compare data structures**.
- Improve your knowledge of basic and **complex data structures (Hashing, Balanced Trees, ...)**
- Implement data structures to **solve real computing problems**.

Recommended:

- **Before** each lecture:
 - read the selected chapters
 - read the lecture slides (from past weeks)
 - note any questions (for next lecture)

Recommended:

- **During** each lecture:
 - participate **actively**
 - do check-in activities or any other participation activities
 - be quiet and attentive
 - stop me anytime if you have questions

Recommended:

- **After** each lecture:
 - Join **Office Hours** if you need help with **anything**
 - Review the slides and complete your notes if needed
 - Do the readings again (current week and next week)

What will we discuss?

Data Structures

- Dynamic Lists
- Linked Lists
- Stacks and Queues
- Trees
- Graphs
- And more...

Application of Prior Programming Concepts

- Generics
- Iteration
- Recursion

Why is this **course** important?

03- Syllabus, Tentative Schedule, and Administrative



About the course...

Read the syllabus carefully

Let's go through the key points

Where to find the schedule?

How is participation evaluated?

Extra-credit?

About the course...

Textbooks

- Required: Mark Allen Weiss, Data Structures & Problem Solving Using Java, 4th ed., Addison-Wesley, 2010.
- References: Frank M. Carrano and Timothy M. Henery, Data Structures & Abstractions with Java, 5th ed., Pearson

It is assumed that students read the scheduled sections of the required book prior to each lecture.

About the course...

Piazza

- Not quite like previous programming courses.
- Use Piazza to discuss any programming topics between you (without sharing code or implementation details).
- TAs will assist in the discussion if necessary.
- Use private posts in case you need to show code or discuss implementation details.
- Use Email only in very exceptional situations (see syllabus).

About the course...

Piazza vs Canvas vs Gradescope vs PollEv

- Piazza: for discussion, assignment descriptions, announcements.
- Canvas: Mainly for Grades posting
- Gradescope: You will submit your work there
- PollEv: For eventual in-class and survey activities



About the course...

Office Hours

See Piazza more accurate info.

Browser address bar: piazza.com/gmu/fall2025/cs310/resources

Navigation bar: PIAZZA CS 310 Q & A Resources Statistics Manage Class Archange Giscard Destine

Course Information

☒ Manually sort using ☐ Sort on: - -

Course Information	Actions
<input type="checkbox"/> Syllabus	<input type="checkbox"/> Edit <input type="checkbox"/> Post a note <input type="checkbox"/> Update link <input type="checkbox"/> Delete
<input type="checkbox"/> Tentative Schedule	<input type="checkbox"/> Edit <input type="checkbox"/> Post a note <input type="checkbox"/> Update file <input type="checkbox"/> Delete

☐ Add Links ☐ Add Files

General Resources

☒ Manually sort using ☐ Sort on: - -

General Resources	Actions
<input type="checkbox"/> Transfer Student Help Video 1	<input type="checkbox"/> Edit <input type="checkbox"/> Post a note <input type="checkbox"/> Update link <input type="checkbox"/> Delete

About the course...

Grading

3.1 Assessment and Grades (click to expand)

Category	Percent
Coding Warm-Up	1%
Programming Assignments (4 projects)	36%
Participation	5%
Midterm Exam	25%
Final Exam	33%
Additional Exam Requirement: You must pass the final exam (60% or higher on both the theoretical and the programming parts)	

The following will be applied *without* rounding:

- A+ ($\geq 98.0\%$) A ($\geq 92.0\%$) A- ($\geq 90.0\%$)
- B+ ($\geq 88.0\%$) B ($\geq 82.0\%$) B- ($\geq 80.0\%$)
- C+ ($\geq 78.0\%$) C ($\geq 72.0\%$) C- ($\geq 70.0\%$)
- D ($\geq 60.0\%$)
- F ($< 60.0\%$)

About the course...

Programming Assignments

36% of your final grade

Project_1: 9%

Project_2: 9%

Project_3: 9%

Project_4: 9%

About the course...

Programming Assignments Deadlines

Late submission at most 48h after the deadline.

1s - 1 day late: - 10 points

1day 1s - 2 day late: - 20 points

Read the syllabus carefully about the late penalty policy.

About the course...

Participation

Very low percentage: 5%.... However ...

It is a Key factor for the success in this course

Attend to lecture, participate in activities, piazza discussions,
do the readings

And some extra-credit opportunities

About the course...

Participation

Components:

- (Soft) Attendance
- Quizzes

Before each lecture

Readings (see detailed schedule)

Review previous slides (prepare questions)

Post any questions on Piazza or during lecture

It is a 15-week course, do not keep your questions for later

During the lecture

Feel free to raise hand and ask questions anytime (use **PollEv** if **you prefer**).

Participate actively in coding activities (you should expect to become good programmers after CS310)

What are the prerequisites?

Grade C in CS211 ...

Coding Warm-Up (doable in 1 day) is checking this requirement

Due **Sept 5, 11:59pm**

Skills needed:

- Generics
- Writing small Java classes
- Good style and comments



In case you forget some of those concepts

Coding Warm-Up provides some guidance

Some general resources under **Piazza/Resources**

If you need help:

- Use **Piazza** (private post if questions contain code or implementation details).
- Join **Office Hours**
- Learn about **Academic Standard**

Introduction to Data Structures

- 1. What is a **Data Structure**?
- 2. **ADT** vs **DS**
- 3. *Quick Overview of common DS*

What is a Data Structure

From CS 211...

Data type... a group of objects (values) and operations we can perform on these values...
... defined within a specific programming language.

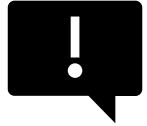
In CS 310...

We will talk a lot about **Abstract Data Type (ADT)**.

ADT : **Specification** for a group of values and the operations on those values.

... **more conceptual** and is independent of any programming language.

What is a Data Structure?



“A data structure is a **representation of data** and the **operations** allowed on the data” (Weiss, Chapter 6).



Other textbooks will also refer to these (data/fields and operations on them) as **Abstract Data Types** (ADTs) (Lafore).

ADT vs DS

A Data Structure is just the **implementation** of an **ADT** within a **Programming Language**.

For this Data Structures course, we will use **Java**.

It is important that you master the key concepts and programming skills that you will need (learnt from CS211).

It you need revisions of those concepts; some resources are provided ...



Why learning Data Structures?

Think about it...

You want to store a large number of Bank clients:

How to store those records efficiently?

You will need to search those clients by ID or by Name...

How to do that efficiently?

What if now you want to add a new client?

Why learning Data Structures?

Think about it...

You want to store a large number of Bank clients:

How to **store** those records **efficiently**?

You will need to **search** those clients by ID or by Name...

How to do that **efficiently**?

What if now you want to **add** a new client?

Would this approach work for 10,000 clients?

Why learning Data Structures?

What about 10,000,000 clients ?

Would this approach still work?

So many alternative solutions...

We expect that you will be able to make the right decision(s) in terms of Data Structures after completing this course.

Why learning Data Structures?

Data Structures

- (Arrays)
- Dynamic Lists
- Linked Lists
- Stacks and Queues
- Trees
- Graphs
- And more...

Quick Overview of Common Data Structures

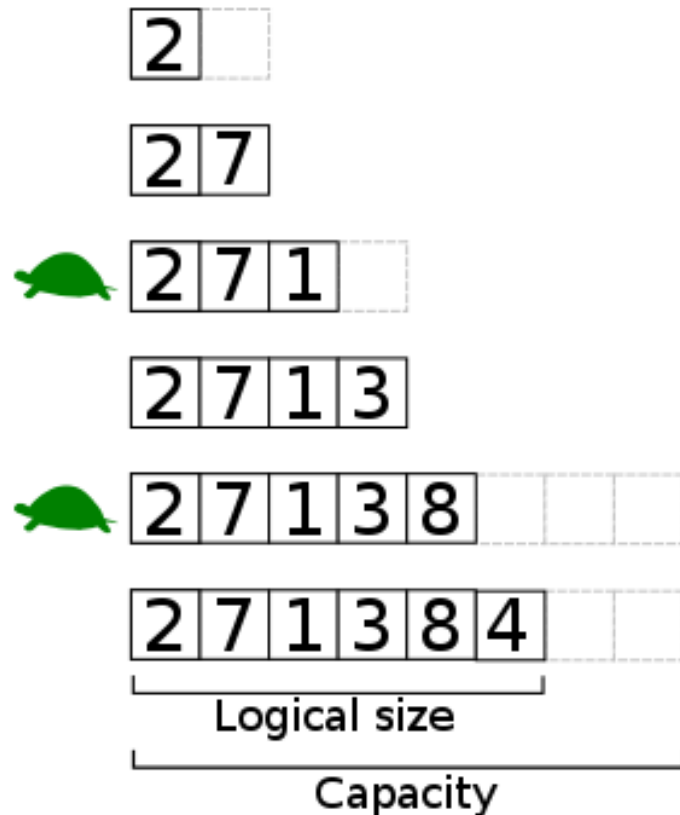
Arrays...

Pros and Cons

Quick Overview of common DS

Dynamic Array Lists ...

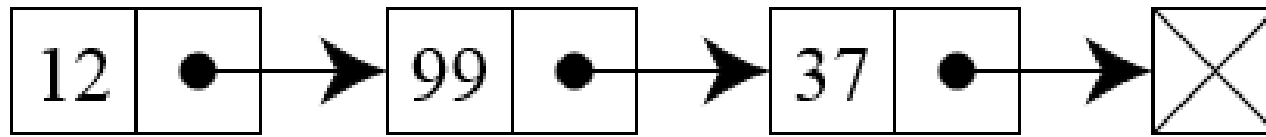
Capacity can grow (and/or shrink to adjust to the number of items)



Pros and Cons ?

Quick Overview of common DS

Linked Lists



Pros and Cons ?



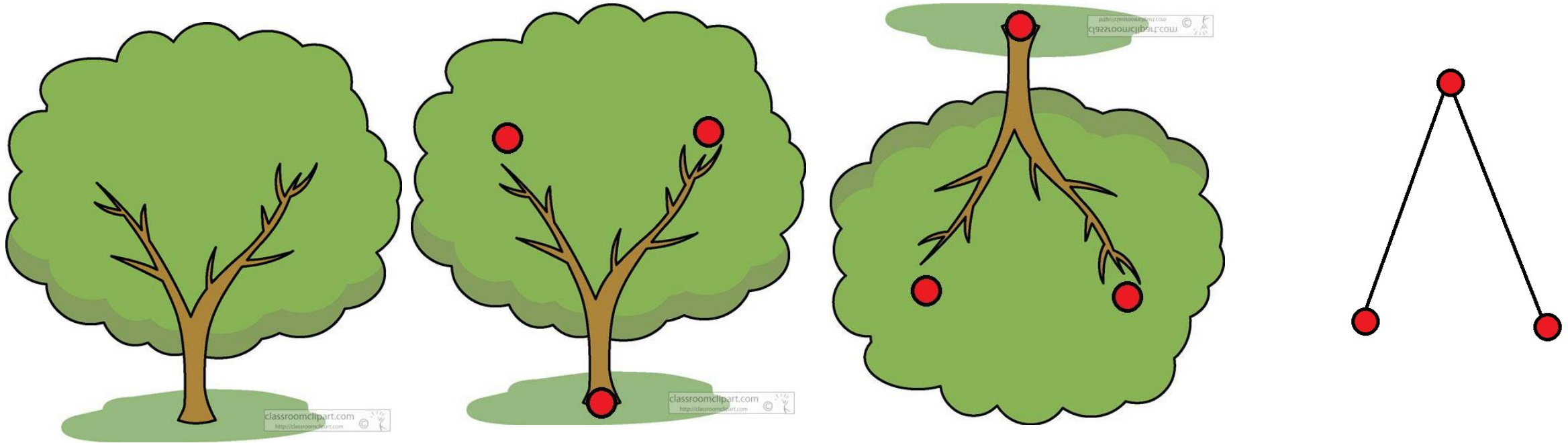
Quick Overview of common DS

From List to Tree... Why?

Hint... Binary Search performance
(we will discuss **time complexity** later)

Quick Overview of common DS

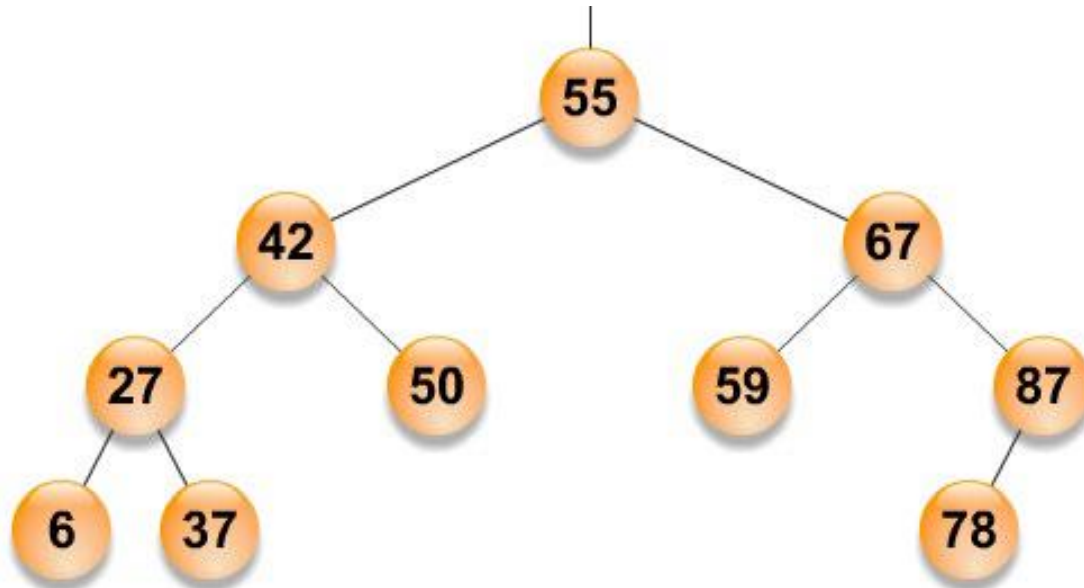
Tree-



Quick Overview of common DS

Tree-

And different type of trees... Why?

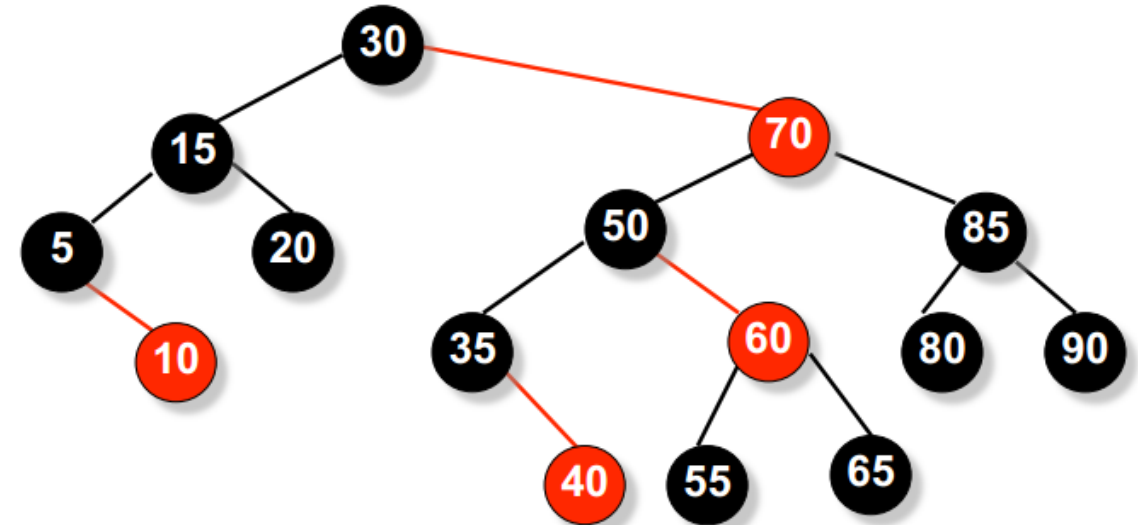
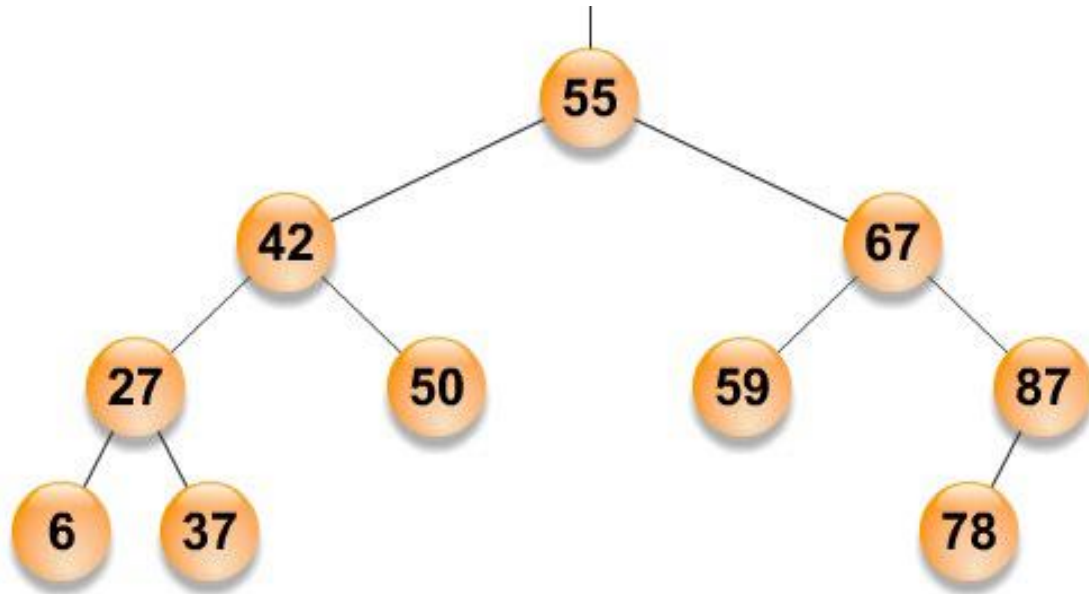


Quick Overview of common DS

Tree-

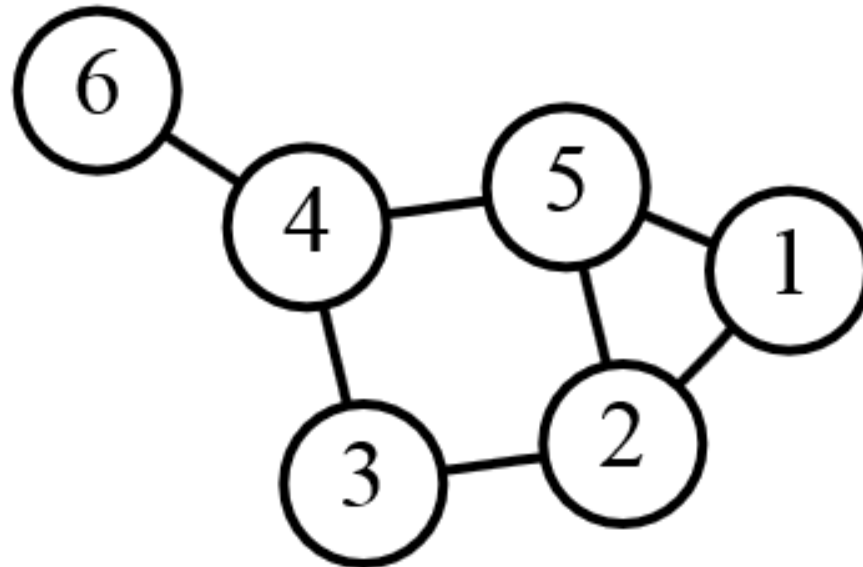
And different type of trees... Why?

Quick Overview of common DS



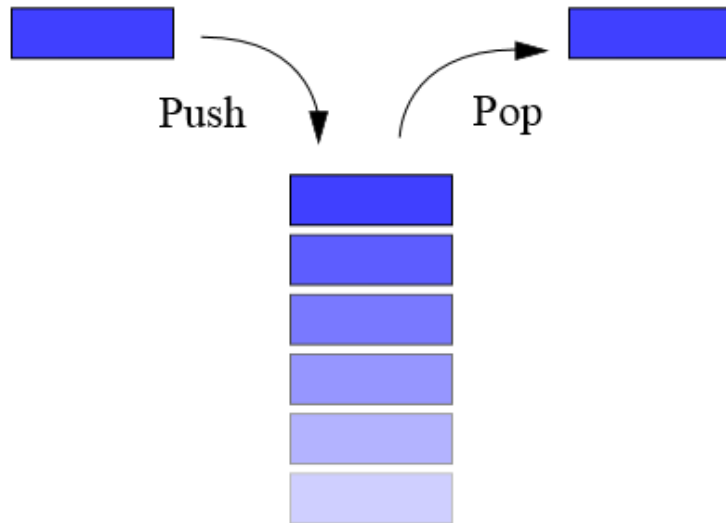
Quick Overview of common DS

Graphs-



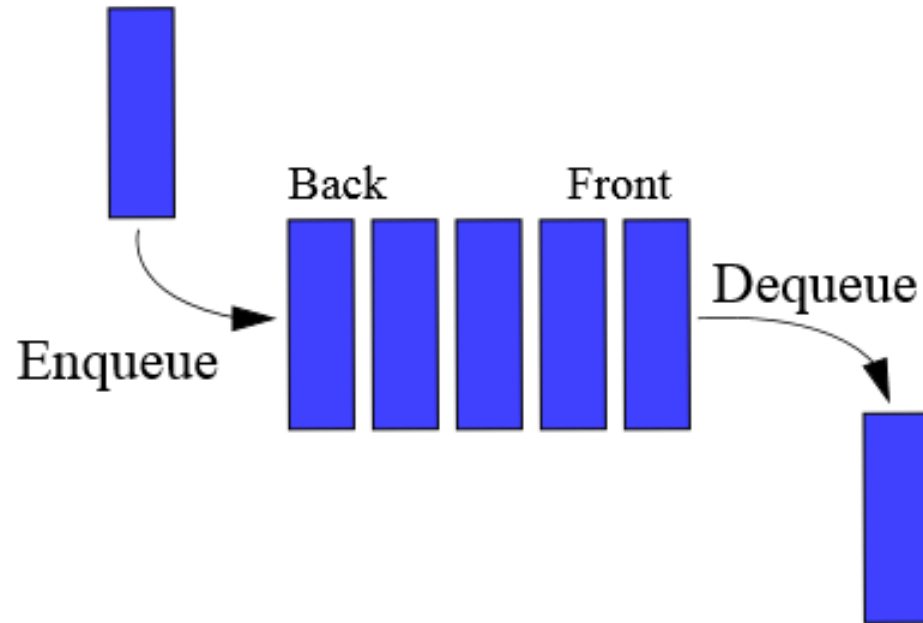
Quick Overview of common DS

Stack -



Quick Overview of common DS

Queue -





Quick Overview of common DS

And more:

Priority Queues

Sets – { 1, 10, 5 }

Maps – { “fred”: 1, “alex”: 2 }

We will focus on...

How does the DS work?

What are the benefits / trade-offs?

We will also use...

Prior Programming Concepts

- Generics
- Recursion



Before we start discussing DS

*How many of you are **Transfer Students**?*

*Should we review **Generics**?*

*Should we review **Recursive Methods**?*

Generic Classes

Basically: A better way to reuse your code with different data types

In Java...

```
class ClassName<X, Y> { ... }
```



and

```
ClassName<Integer, Banana> x = new ClassName<>();
```

Recursion

Call a **function** from inside the same **function**

Key components

- recursive case
 - when to repeat the steps
- base case
 - when to stop

```
public static long factorial(int n) {  
    if (n == 0 || n == 1)  
    {  
        return 1;  
    }  
    else  
    {  
        return n * factorial(n - 1);  
    }  
}
```



Coding Warm-Up

Where to find Project description ?

What if you need help? **(Do not just show your code...)**

What are **Office Hours** for?

Let's do a quick coding review now.

Java Basics - Quick Review

Almost everything is a class

```
1 public class Program0 {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Welcome to CS310");  
5     }  
6  
7 }
```

Java Basics - Quick Review

A simple program (See Basic0.java)

Defining a **function**

Using **Generics** (See Operation.java... and Program2.java)

Overloading the function (See Program1.java)

Using **Recursions** (See Program3.java)

Proper **comments** and **style** (See Program4.java)

(Quick review demo, code posted on Piazza)

Important for this course: Algorithm Analysis

- Algorithm
 - = how you do things

- Algorithm Analysis
 - = analyze how **well** you do things

- What defines **well**?
 - - Time to complete?
 - - Computer memory required?

Next Lecture

1. More review: **Basics, Generics**
2. **Efficient programming** / Computational Analysis

Reminders:

Work on Coding Warm-Up

Do the readings (for this week / see the schedule)

Fill the Survey (using PollEv)