CSCI 132:
Basic Data Structures and Algorithms

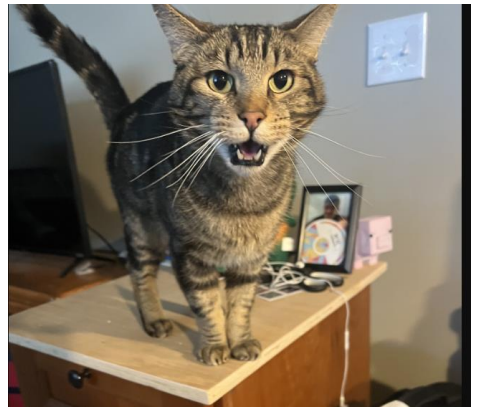
Final Exam Review

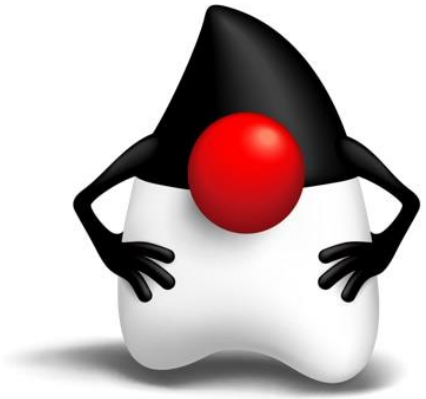
Reese Pearsall & Iliana Castillon

Fall 2024

Announcements

- Program 5 due **Sunday**
- Final Exam on Monday (12/9) at **2:00 PM – 3:50 PM** in our normal classroom
- Rubber duck screenshot due tonight





Meatball and Sasha wish you good luck on your final exams

Final Exam Logistics

Same format/rules as the midterm exam

- You can use notes, your IDE, lecture recordings, previous assignments, java documentation. No external resources

- Bring your laptop if you need

Roughly about the same length as the Midterm, but you have 2 hours this time ☺

8 Parts

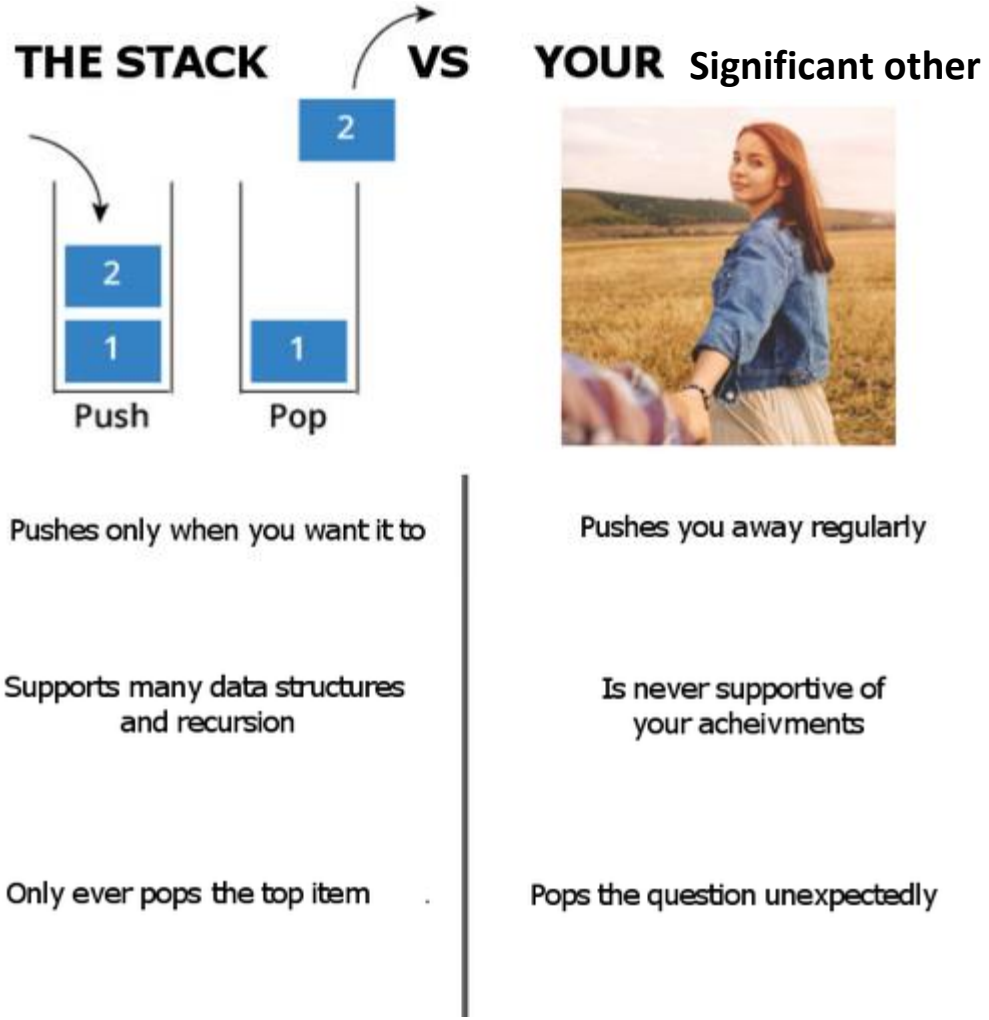
1. Multiple Choice
2. Short Answer
3. Sorting
4. Searching
5. Basic Java Classes
6. Stacks
7. Recursion
8. Queues

Basic Java Class Structure

- Be able to identify/define instance fields and methods
- Write a constructor
- Understand basic Java keywords
- Understand Java reference variables
- Write a basic java method

Stacks

- Be able to understand basic stack methods (push pop peek)
- Given code that utilizes a stack, be able to visualize and illustrate the contents of a stack
- Know the running time of stack operations
- Write code the uses a stack



Queues

- Be able to understand basic queue methods (enqueue dequeue peek)
- Given code that utilizes a queue, be able to visualize and illustrate the contents of a stack
- Know the running time of queue operations
- Write code the uses a queue

Recursion

- Given a basic recursion function, derive the output and number of recursive calls made
- Understand how to calculate the running time of a recursive algorithm
- Understand limitations/benefits of recursion

Sorting

- Bubble sort, selection sort, merge sort, quick sort
- Be able to describe/illustrate the steps of these sorting algorithms
- Know the running time for each sorting algorithm
- Know which ones are efficient/not efficient

Searching

- Understand the differences between linear search and binary search
- Understand the running times of those algorithms
- Be able to look at code for linear search and binary search and understand what is happening

Short Answer

- Basic Java Classes, Class Structure, Methods, Operations, if statements, loops, OOP
- Basic Linked Lists
- Big-O Notation, How to determine running time of an algorithm
- Stacks
- Queues
- Bubble Sort
- Selection Sort
- Merge Sort
- Quick Sort
- Linear Search/Binary Search
- Recursion

Multiple Choice

- Basic Java Classes, Class Structure, Methods, Operations, if statements, loops, OOP
- Basic Linked Lists
- Big-O Notation, How to determine running time of an algorithm
- Stacks
- Queues
- Bubble Sort
- Selection Sort
- Merge Sort
- Quick Sort
- Linear Search/Binary Search
- Recursion

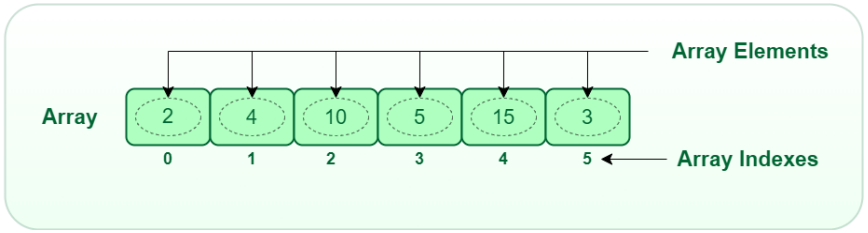
Final Exam Study Guide

Course Goals

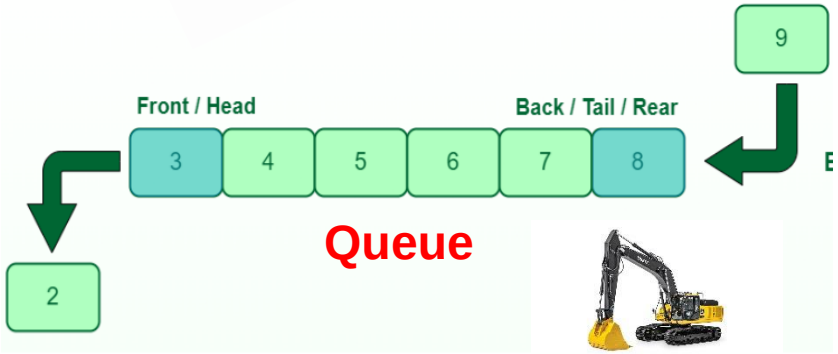
- Design and Implement programs of simple and moderate complexity in Java
- Explain the concept of an ADT *(meh)*
- Understand and implement basic data structures: Linked lists, stacks, and queues
- Given a simple algorithm, determine the time complexity using Big-O notation
- Understand basic searching and sorting algorithms and their runtime
- Understand how recursion works, be able to analyze recursion runtime, and be able to implement recursion in a program
- Be able to debug programs and become an independent problem solver

Takeaways

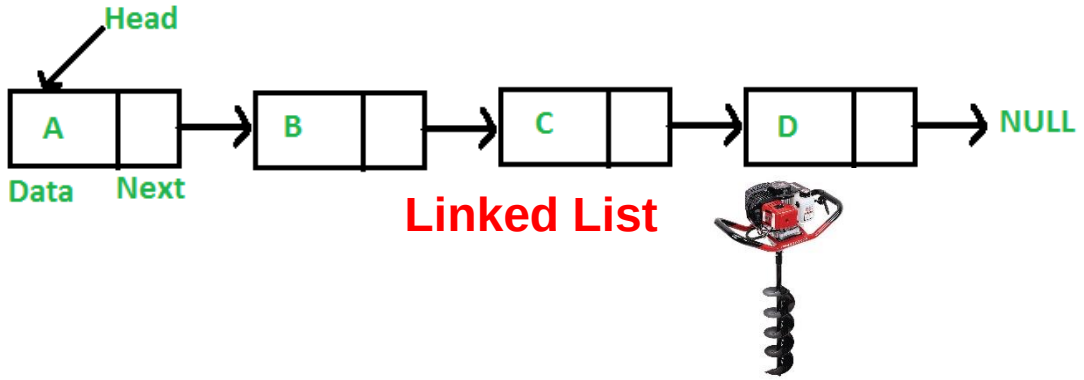
We have different data structures that handle data differently. There are **tradeoffs** between using these data structures



Arrays

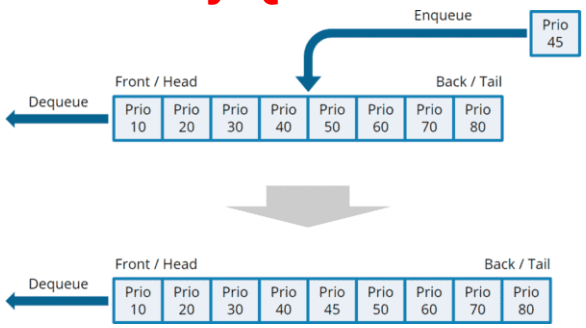


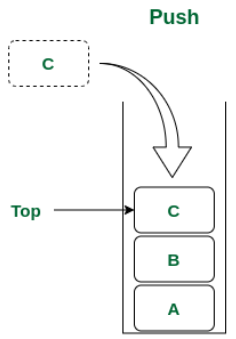
Queue

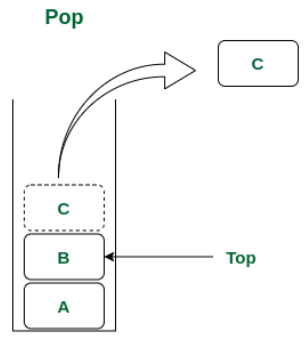


Linked List

Priority Queue







Stack

Given a problem, you should be able to identify a good candidate for a data structure and provide a justification

Takeaways

- There be many different types of algorithms. Every algorithm has a running time, which is important to be aware of
- The algorithm you select is important. It can be the difference between your program finishing in 6 seconds, or you program *never finishing at all*
- The data structure you select is important. When deciding which data structure to use, you should have a reason to back it up
- We have methods for measuring the efficiency of some algorithm (big-O notation).
- When you write an algorithm, you should be able to broadly describe the effectiveness and efficiency of it

Our Goals for you

Get you comfortable with writing basic Java programs

Give you a good toolset that can help you solve a variety of problems (Data Structures)

Give you techniques and methods for solving a variety of problems (Algorithms)

Give you the skills to analyze the algorithms that you write (Big-O notation)

Thank You!

This class has been fun to teach for us to teach. I understand that there were certain parts that were not very exciting.

We hope you enjoyed this class, and we hope the stuff you learned will be helpful in your career/future classes

If we can be of assistance to you for anything in the future (reference, advising, support), please let me know!

There is a very good chance you will have in our classes in the future

Connect with us on LinkedIn!

Reese Pearsall (He/Him)
Instructor at Montana State University
Bozeman, Montana, United States · Contact info

Iliana Castillon · 1st
Instructor at Montana State University
Bozeman, Montana, United States · Contact info