Unit 4 Notes

11/13/23 - More Recursion

Recursion Review

- a way to `repeat` code without loops
- Methods that call themselves
- Recursive methods have
    - A base case (condition to stop)
    - Recursive call
    - Return values (good design)

The Memory Stack

- Math factorials
    - N = 6, factorial is $1 \times 2 \times 3 \times 4 \times 5 \times 6$
- When calling methods
    - The method is pushed onto the memory stack
    - Removed when done
- This causes the following to happen in memory
    - You will cover this more in CS250 and 220

Overview

- Recursion
    - When to use it?
        - When you have limited paths to follow
        - When you don’t know your loop depth

        - When your data is already set up a tree
    - You will come across it again - CS165
    - Always remember your base case!
- For example let’s consider the following sequence of numbers
    - `[0,1,2,3,4,5,6,7,8,9,10]`
    - Numbers are ordered
    - We could structure this number line as a “tree”

11/27/23 - Arrays 2D

Arrays - Reminder

- Ways to store
    - Variables in order
    - Index from `0...N`
- Arrays are
    - A type themselves
    - The value of the array
        - Reference to memory location
    - `Array.length` gives us the total memory allocated
- Arrays can
    - Be any size - as long as you allocate it
    - Store any valid type
        - Primitives and objects

Easy Access with For-Each

- For each loops
    - Specialized for loops
    - Perfect for an array or other collections
- Loops through every value
    - Stores it in a temp variable
- Same as some very common for-loops

Arrays and Objects?

- Primitives - values are stored
- Objects - references to values

```
Box[] values = new Box[10];
MyObject[] values = new MyObject[5];
```

- Can you have an array of arrays
    - Yes
- Arrays have type
    - Anything with type can be an array

2D Arrays

- Array of arrays
- Same type
- Very common
    - So much so we have a shorthand notation

```
int[][] arr2d = {{1,2,3},{6,7,8},{12,13,14}};
```

2D Arrays - Irregular / Ragged Arrays

- You can have arrays of variable length within an array

- Those are called “irregular or ragged” arrays

11/29/23 - Advanced Topics: Collections

ArrayLists

- Part of the Java Collections library
- Assumes default naming conventions
    - Done through interfaces and abstract classes
- `.add(Type)`
- `.remove(location)`
- `.size()`
- Is ArrayList always the best to use?
    - what happens if it is *very* large?
    - Hard to find continuous memory in order
    - Causes actions to slow down
- Introducing Data Structures (CS165)

LinkedList

- Think about a chain
- Each link connects to the next
- Linked Lists
    - Connect objects to the next
    - But don’t want to worry about it all being ordered in memory
- If you know the next, they can be anywhere
- Pros
    - memory efficient

- Cons
    - What if a link is broken?
    - Can you easily jump to the middle? No!
- Also a foundation of blockchain

LinkedList vs. ArrayList

- A LinkedList typically provides faster element insertion and removal at the list's end (and middle if using ListIterator)
    - LinkedList methods with index parameters, such as `get()` or `set()`, cause the list to be traversed from the first element to the specified element each time the method is called. Thus, using the LinkedLists' `get()` or `set()` methods within a loop that iterates through all list elements is inefficient
- ArrayList offers faster positional access with indices
    - It maintains a based system for its elements as it uses an array data structure implicitly which makes it faster to search an element in the list

LinkedList - Practical Examples

- Image viewer - Previous and next images are lined and can be accessed by the next and previous buttons
- Previous and next page in a web browser - We can access the previous and next URL searched in a web browser by pressing the back and next buttons since they are linked as a linked list
- Music player - Songs in the music player are linked to the previous and next songs. So you can play songs either from the start or end of the list

Map

- What if you had key-value pairs?
- Example: your address points to your house
    - Does a book of addresses, store all the information about your house?
    - Or simply the address, that can get the info?

- Introducing maps
    - Pairs keys to values
    - Keys need to be unique
- Some uses: database indexing, network routing

Why does it matter?

- Different data structures
    - affects speed, memory, and storage
    - Important for all fields
        - Biology - large datasets
        - Graphics - speed is needed
        - Cybersecurity - processing serialized information over networks
- If you interview at FAANG
    - They often give you a tech quiz
    - Most of what is on that quiz - you learn in CS 165
- Take CS 165, it provides a major programming foundation.
