

Week 01 Notes

Prof Bill - Jan 2020

Week 01 is **Mon Jan 6 - Fri Jan 10**

Week 1 notes on:

- A. CSC 210 in 15 minutes or less
- B. Array, ArrayList, Linked List
- C. Java intro

thanks... yow, bill

Getting to know you:

- Really random: www.random.org
- Questions: www.signupgenius.com/groups/getting-to-know-you-questions.cfm

Our books:

- ❖ Morin, Data structures book: opendatastructures.org/
- ❖ Sedgewick, Java book: introcs.cs.princeton.edu/java/home

Culture alert! The Russian ballet is right here at Noctrl on Sat Feb 8.
I've seen them here before. It is absolutely incredible. Must see!

Russian National Ballet

finearts.northcentralcollege.edu/event/russian-national-ballet-presents-cinderella

A. CSC 210 in 15 minutes or less...

Why are we here?

Example: Build-a-browser

Given URL, get the IP address; example: URL=noctrl.edu, IP=192.160.165.164

Store as array. Search for URL.

Works for 10 websites. What about 10,000,000,000?

Do real browsers get slower as the number of websites increases?

Array

Fixed number of cells, adjacent in memory.

```
int[] example = new int[10];
```

Operations: Add to end; Add to beginning; Insert; Search; Remove

Advantage: easy, fast. Disadvantage: max size restriction

Linked List

Self-referential node. Flavors: singly-linked, doubly-linked, head, tail

Operations: Add to end, Add to beginning, Insert, Search, Remove

Advantage: No max size. Intuitive.

Disadvantage: No $O(1)$ indexing into the list, garbage collecting nodes.

ArrayList

EZ rule: If we blowout our array size, then make it bigger.

Removes the max size disadvantage of an array.

Hash table

Goal: I'd like to get the array $O(1)$ search by index for everything, not just `array[i]`.

Problem: But not everything is an integer/indexable. Like a name: "Prof Bill"

Solution: **Magic**. Create a "hash function" that turns "Prof Bill" into an integer.

Hash tables are so important...

- Older - C has no support, so you built your own
- Medium - Java has support in its standard class library: `HashMap`
- Newer - Javascript and Ruby have hashes incorporated into the language

Javascript: every object is a hash, where attributes and methods are hashed.

medium.com/javascript-in-plain-english/algorithm-in-javascript-hash-table-7b0464d2b81b

Ruby:

```
fun210 = { "one" => "St Clair", "two" => "Krieger", "three" => "Kardaras"
}
puts fun210["two"] # "Krieger"
```

Algorithm Analysis

Handout: Big-Oh

Wrap

The BIG 4 are key: Array, Linked List, ArrayList, Hash table (**magic**)

Algorithm analysis is easy and useful, not mathy.

210 is a fun mix of theory and practicality. Let's code!

210 is less important now than before (gasp!): Moore's Law, database

B. Arrays, ArrayLists, Linked Lists

The BIG three most basic data structures.

Arrays, ArrayList

Text: Morin 2.1-2.4

Big-Oh:

- $\text{get/set}(i) = O(1)$, constant
- $\text{insert}(i), \text{remove}(i) = O(n)$, linear # because you have to move elements in array

Array is a contiguous block of memory; size is set when created

Arrays cannot expand or shrink after being created; error if you exceed array size

Array often used to implement Stack or Queue ADT...which we'll later/separately.

ArrayList is an array that resizes itself if necessary.

Common approach: double size each time you resize.

Key method to do this without looping: **System.arraycopy**

ArrayList is a very popular class in the **Java Collections Framework, JCF**.

beginnersbook.com/java-collections-tutorials

Linked Lists

Text: Morin 3.1-3.2

List is comprised of nodes.

- **Singly-linked list**: each node has a next pointer.
- **Doubly-linked list**: each node has a next and prev pointer.

Lists maintains a pointer to the **head** and **tail** nodes.

/ some nice pseudo-code for lists in the Morin text */*

Hash tables

Text: Morin 5.1-5.2

Text has get/put(x). x is really [key, value].

Basic operations:

- put(K key, V value): hash key into an integer; then store [key,value] at array[hash]
- get(K key): hash key into integer; return object at array[hash] or null if not found

Two most common ways to handle collisions:

- **Chaining**: linked list of objects at each hash location, array[i]
- **Linear probing**: on collision, put object in next available array[i++] spot

Linear probing is also called open addressing.

All Java objects have a **hashCode()** method.

JCF has (very popular) **HashMap**.

docs.oracle.com/javase/8/docs/api/java/util/HashMap.html

Woof, that's a really bad Morin chapter; Sedgewick Algos is a little better

algs4.cs.princeton.edu/34hash

/* my separate/extensive notes on hash tables coming next week. So important! */

C. Java Intro

Java vs. Python

Y'all know **Python**. How does it compare to Java? Google it up: java vs python
Good site with some code snippets to compare, raygun.com/blog/java-vs-python

My two cents:

- ❖ Java is more verbose, static typing; Python more fun (?)
- ❖ Java uses curly braces; Python uses indentation
- ❖ Python libraries are giving it strong mojo
- ❖ I've seen this a lot: Python is easier for beginners
- ❖ Many colleges (including Noctrl CSC) are moving from Java to Python for programming 101

Java intro

Text: Sedgewick Java 1.1-1.5

Sedgewick Java book:

- pretty good; better than Morin; copyright 2017
- weak on generics, JFC, Javadoc, etc
- Question: use his libraries (StdIn, StdOut, StdDraw)? These are not a part of the standard Java library

Java online

Compiling and running Java in a web page is fun.
I googled. I played. I liked this one the best. So far.
www.onlinegdb.com/online_java_compiler