Exam 3 Review

Data Structures

COMP128 Data Structures

Learning Goals We Covered So Far

- Arrays
- Time Complexity Analysis
- Stacks
- Queues
- Lists
- **Maps**
- **Tree**
- **Sorting**
- **Data Structure Choices**

Take-home Format

- Allowed resources is similar (need to sign a honor statement at the end)
- Exam will be posted at **6pm this Thursday** (Apr. 14th) on Moodle (pdf file) (no class this Friday)
- Complete and turn in the paper before **6pm this Friday (Apr. 15th)**
- I will be around in my office (OLRI 158) on Friday, and if I am not around, you can slide the paper under my office door

Reviewing Data Structures

For each data structure:

- How to work with each data structure
    - Add
    - Remove
    - Access
    - Iterate
- How to implement this data structures
    - Different approach
    - Time complexity

Example: Map

For each data structure:

- How to work with each data structure
  - Add
  - Remove
  - Access
  - Iterate
- How to implement this data structures
  - Different approach
  - Time complexity

Example: Map

For each data structure:

- **How to work with each data structure**
  - **Add:** put(key, value)
  - **Remove:** remove(key)
  - **Access:** get(key)
  - **Iterate:** .keySet(), entrySet()
- How to implement this data structures
  - Different approach
  - Time complexity

Example Problem: Map

In an online app store such as google play or the apple store you will often use customer ratings as a way of measuring which apps are good and which are not. Suppose you are building software for such an app store. In this software ratings are a whole number between 1 and 5 (1, 2, 3, 4, or 5). **To store these ratings we will use a map**. The **keys** of the map are Application objects. The **values** in the map are **linked lists of numeric ratings**. **Write a method named computeAverages which takes the application map and returns a new map with each app's average rating.**

Example Problem: Data Structure Choices

Suppose that you are building an application in which you wish to store a collection of objects and associated data. Your application has the following features:

- The code will dynamically insert and remove objects over time as users interact with the application.
- The objects in the collection will occasionally need to be printed in sorted order.

What data structures (Java Collections Framework) would suit this purpose and why?

Discuss Problem: Tree

This question concerns binary trees.

(a) Under what circumstance might arrays be used to implement a binary tree?

(b) Why would using an array implementation of a binary tree for the above circumstance be beneficial?

(c) What is the disadvantage of using an array implementation for some binary trees?