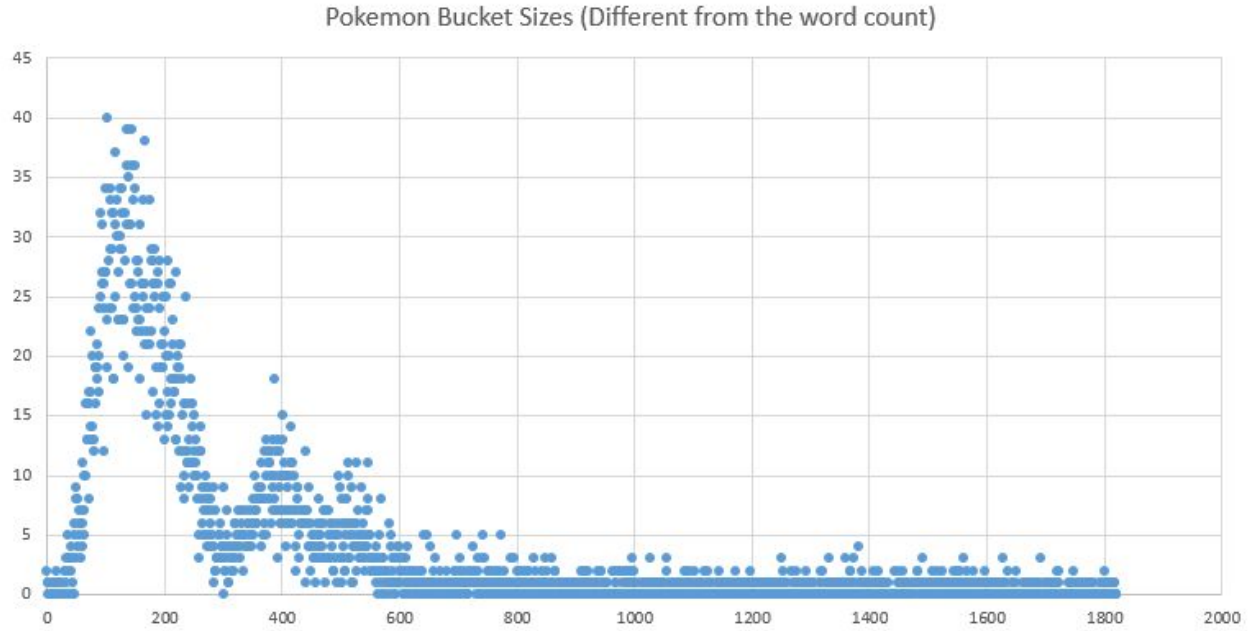
Comp 2071 - Data Structures
Lab 5 - Hash Tables

Group: 11
William Lawrence - Code, Presentation
Tom Plano - Code
Cliff Anderson - Code
Artur Janowiec - Code, Write up

Pokemon Bucket Sizes



ADT

- Nodes use String and AtomicInteger data type
- Constructor is passed initial size
  - Checks greater than 0 and is prime/makes it prime
- Put method is passed a string and atomicInteger
  - Checks if node with string reference already exists, adds new atomicInteger
  - Otherwise creates new node and adds node to hash table
  - Checks to see if array needs resizing
- Get method is passed string key
  - Looks through hashtable until string is found or end
  - Returns stored data or exception

Application

- Asks user for table size then creates table
- Reads in all lines from given text files
- Put strings into table
- Prints metrics to file
- Prints data to file
- Repeats for each file

Issues and Solutions

- Will most of the code your write can be completely re-written
- Tom and Cliff had an issue with constructing a prime number checker from scratch, as it flagged many numbers incorrectly at first
- Artur didn’t have any issues.

Take-aways

- Will learned how to make a hash table
- Tom and Cliff learned that it is better to be efficient and look up code for things that definitely already exist, instead of wasting time trying to be clever and come up with the same code on their own
- Cliff also learned how to efficiently read in files, increasing speed by ~30x
- Artur learned the difference between iterators in data structures like ArrayList vs data structures in HashMap.