Project Part 3 – HashMap

This assignment is a simple custom hash map labeled ElijahWhiteSimpleHashMap.java. The whole goal of this assignment or project I feel like was to show of our Data Structure and Algorithms skillset. The hash map I created uses an array of linked lists and each spot holds a list of items and where each item may go depends on the hash function. To determine where in the array to put the item, the dumbHash hash function uses the length of the string version of the key instead. I felt like using a straightforward approach to better understand how hash collisions occur and how the map responds to them. I also included some functions like contains() to see if something is in the map, put() to add an item, and resize() to double the array's size and spread everything out again when it fills up.

Afterwards I created a tester class named ElijahWhiteSimpleHashMapTester in order to thoroughly test the functionality of my map. Large lists of items, such as "Item 0" to "Item 99999," are created by this tester and added to the map. Additionally, it calculates the time required to add everything and verify that everything is included in the map.

A screenshot of a computer

AI-generated content may be incorrect.

Each row displays the memory usage before and after resizing for various dataset sizes, such as 10,000, 50,000, and 100,000 items, as well as the time it took to insert the items and search.  
 My hash function occasionally put too many items in the same bucket, therefore I saw during the tests that adding items worked quickly at first but slowed down as the number of items increased. The efficiency slightly improved when I applied the resize() method since the pieces were distributed more equally.