Results:

|  |  |
| --- | --- |
| Zero-based index | Data Structure |
| 0 | Linked list |
| 1 | Hash Map |
| 2 | Binary Tree |
| 3 | Linked List |
| 4 | Hash Map |

Most of our deductions were made based on testing the average time it took to find an element in a list. The plot is illustrated below:

\*\*\* Note that the indices in the graph above are one-based and correspond to the zero-based collections.

Index 0 and 3 are linear, so they must either be a heap or a linked list.

Index 1, 2, and 4 are too small, so I generated a new plot with only them:

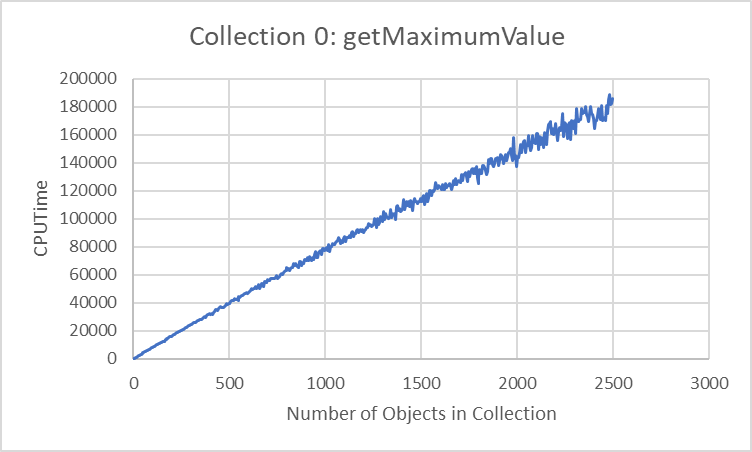
This plot shows that Index 2 is logarithmic, and Index 1 and 4 are constant time and surprisingly identical. Since search is only constant time for hash maps, and since they look the same, I concluded that 1 and 4 are both hash maps. The only data structure we were given that should exhibit logarithmic search time is a binary search tree, so that concluded that index 2 is a BST.

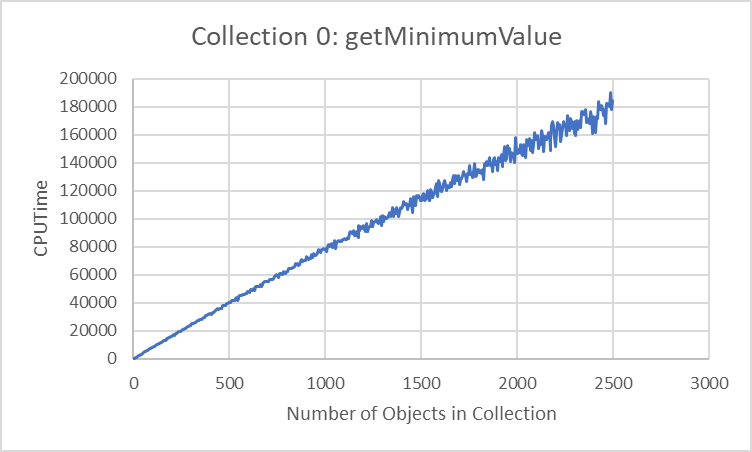
That leaves us to determine whether indices 0 and 3 are heaps or linked list.

To do that, we tested getting the maximum and minimum from each. We expected the following:

* LinkedLists would yield a linear retrieval time
* HashMaps and Max/Min Heaps would retrieve in constant time
* Binary search trees would retrieve in logarithmic time

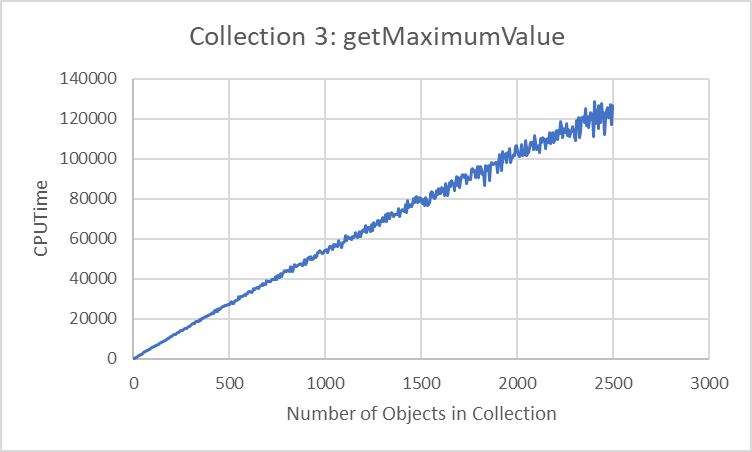
Here are the results of our tests for getting the minimum and maximum value on data structure 0:

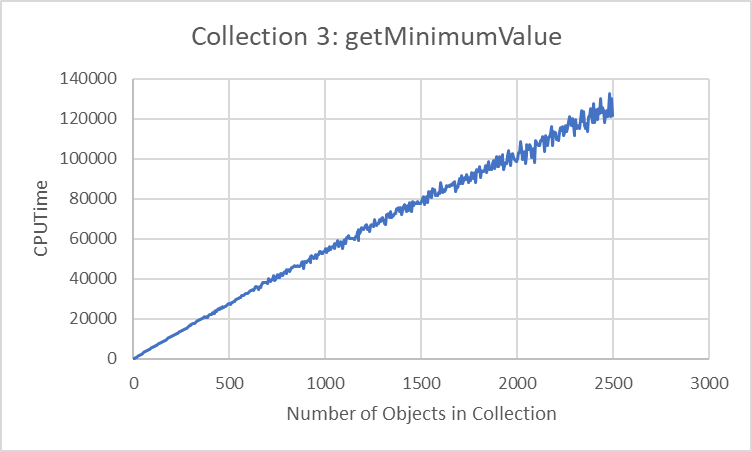
 conclusion: O(n)

conclusion: O(n)

Since both are O(n), these data structures do not exhibit heap behavior. This means the only remaining possibility is a linked list.

Here are the results of our tests for getting the minimum and maximum value on data structure 3:

 conclusion: O(n)

 conclusion: O(n)

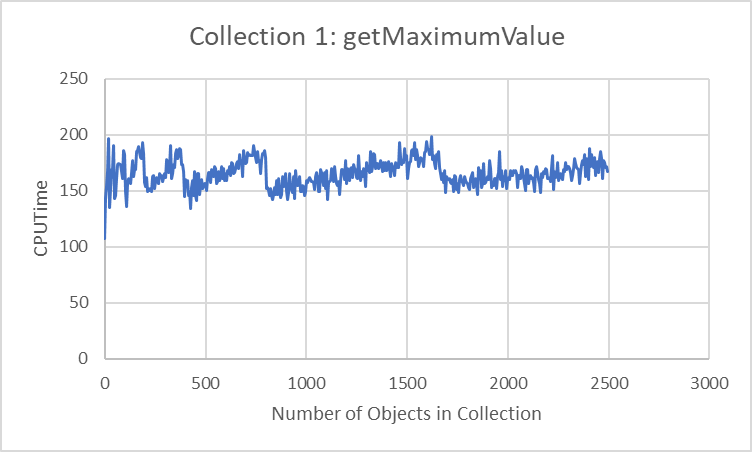
Since both are O(n), these data structures do not exhibit heap behavior. This means the only remaining possibility is a linked list.

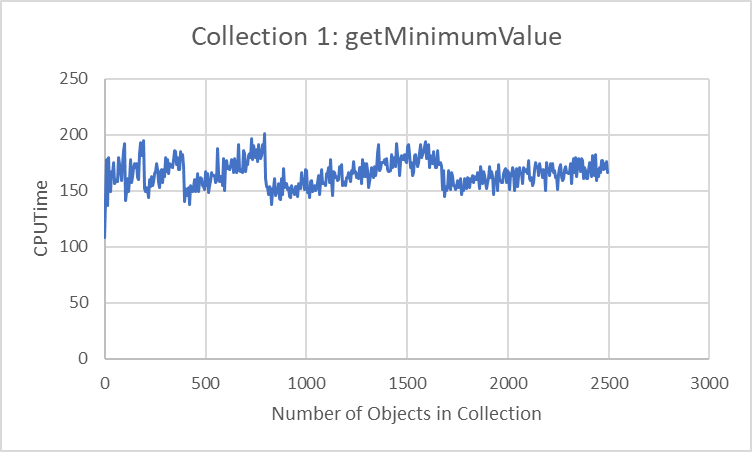
We ran other tests, and to be safe, we made sure that their results matched with our conclusions.

**Collection 0:**

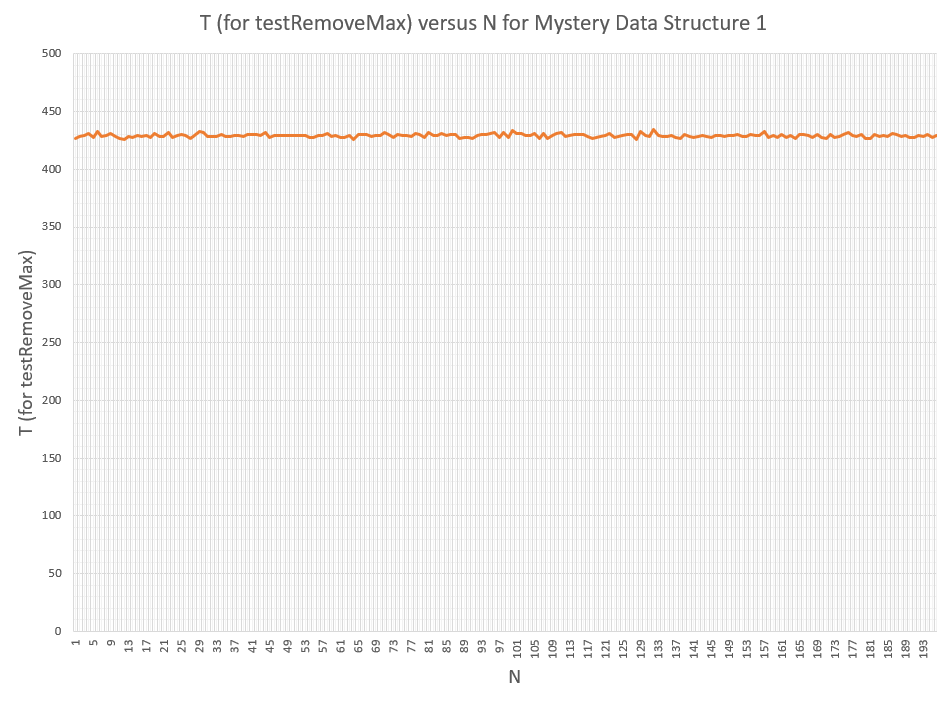
This exhibits linear time for removing the max and indices, which agrees with our original conclusion that it is a linked list.

**Collection 1:**



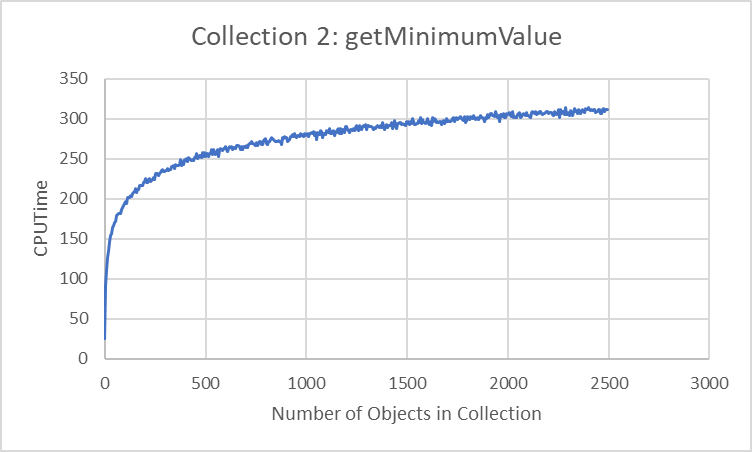


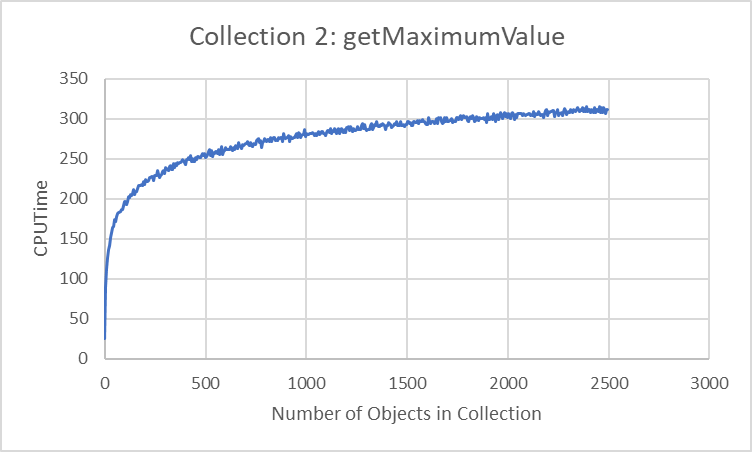
This exhibits constant time for getting the max and the minimum values, which agrees with our original conclusion that it is a hash map.



This also exhibits constant removing time for the maximum value, which would also appear on a hash map.

**Collection 2:**





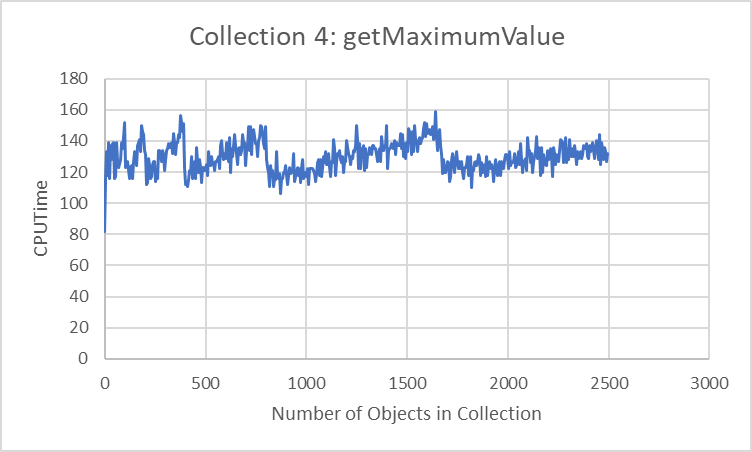
This exhibits logarithmic time for getting the max and the minimum values, which agrees with our original conclusion that it is a BST.

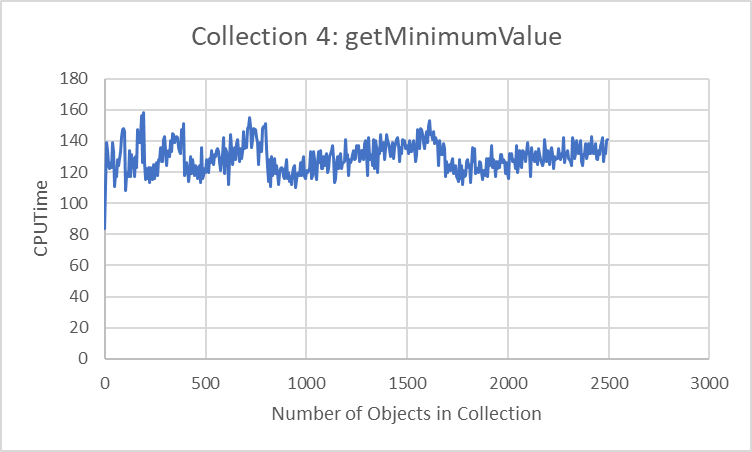
This also exhibits logarithmic removing time for the maximum value, which would also appear on a BST.

**Collection 3:**

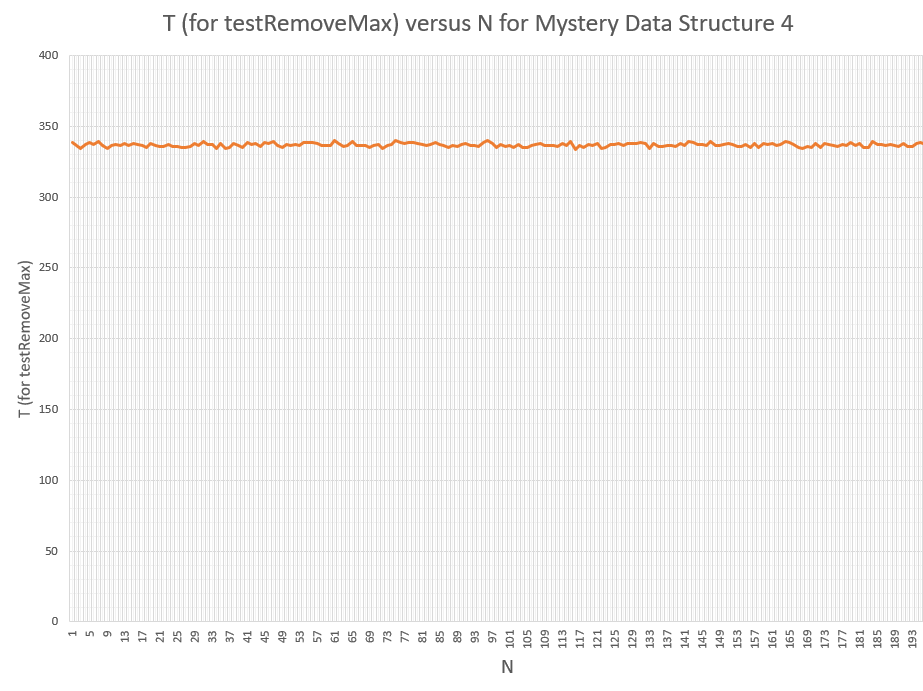
This also exhibits linear removing time for the maximum value, which would also appear on a linked list (which supports out conclusion).

**Collection 4:**





This exhibits constant time for getting the max and the minimum values, which agrees with our original conclusion that it is a hash map.



This also exhibits constant removing time for the maximum value, which would also appear on a hash map.