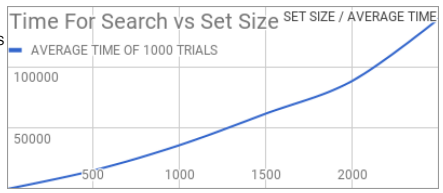
Mikel Matticoli, Harry March - CS210X Project 4 - Timing Test Results

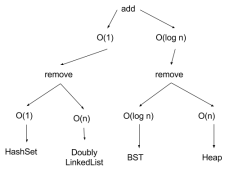
Mystery Data Structure 1

Random Search

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	61
501	15220
1001	36032
1501	61910
2001	88759
2501	139734

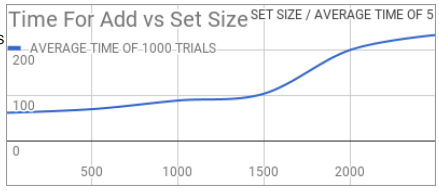


O(N)



Add To Structure

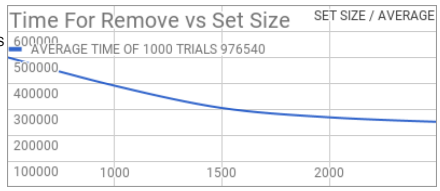
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	61
501	69
1001	88
1501	103
2001	199
2501	232



O(Log N)

Remove From Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	976540
501	499576
1001	389950
1501	304093
2001	268177
2501	250994



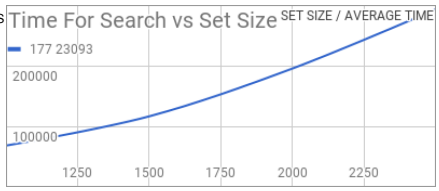
O(1/N)

Since the add operation runs in O(Log N) time and the remove operation is linear with respect to N, this structure is likely some sort of heap. The linearity of the random search's time supports this as the internal representation for a heap would require iteration through N elements to find a particular one.

Mystery Data Structure 2

Random Search

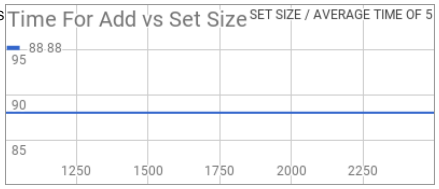
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	177
501	23093
1001	68355
1501	116610
2001	195496
2501	292960



O(N)

Add To Structure

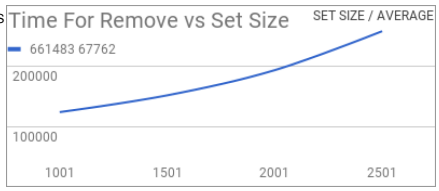
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	88
501	88
1001	88
1501	88
2001	88
2501	88



O(1)

Remove From Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	661483
501	67762
1001	123531
1501	151185
2001	192659
2501	256922



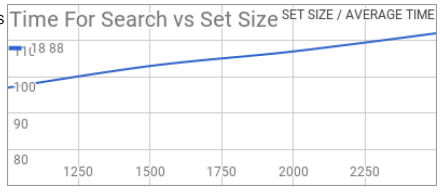
O(N)

Since the runtime for the add operation is very constant, and the timing of the remove function is relatively linear, this structure is a doubly linked list. A search for a random element also appears linear, indicating that this linked list does not store references to each of its nodes.

Mystery Data Structure 3

Random Search

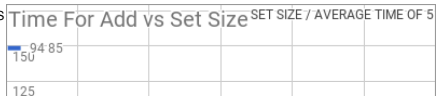
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	18
501	88
1001	97
1501	103
2001	107
2501	112



O(Log N)

Add To Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	94
501	85
1001	89



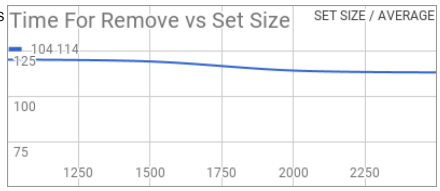
Mikel Matticoli, Harry March - CS210X Project 4 - Timing Test Results

1501	93
2001	97
2501	102

O(1)

Remove From Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	104
501	114
1001	120
1501	119
2001	114
2501	113



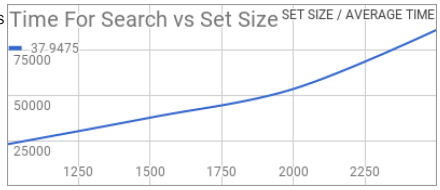
O(1)

Since the add and remove functions for these structures are both relatively constant, this structure is most likely a HashSet. The linear search time supports this, as a search through a HashSet would still require iteration through each of N elements in worst case, despite the fact that the elements are ordered arbitrarily.

Mystery Data Structure 4

Random Search

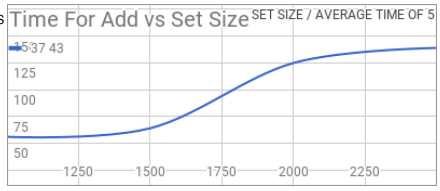
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	37
501	9475
1001	22945
1501	37603
2001	53388
2501	85908



O(N)

Add To Structure

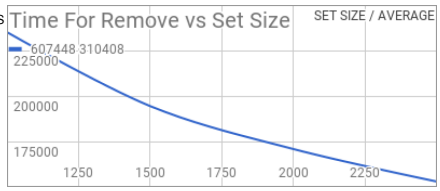
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	37
501	43
1001	56
1501	64
2001	125
2501	139



O(Log N)

Remove From Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	607448
501	310408
1001	235134
1501	194464
2001	170859
2501	153014



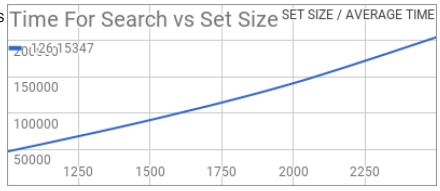
O(1/N)

Since the add function is logarithmic, and the remove function is relatively linear, this structure is most likely a BST. The search function however increases with time, which could indicate that the search algorithm does not take advantage of the fact that the tree is sorted, while the remove function takes advantage of the underlying implementation.

Mystery Data Structure 5

Random Search

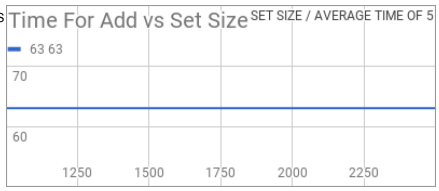
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	126
501	15347
1001	47287
1501	90695
2001	141309
2501	204773



O(N)

Add To Structure

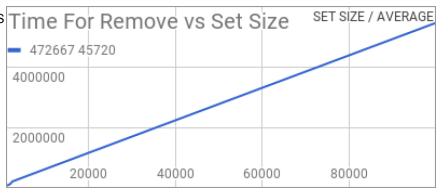
SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	63
501	63
1001	63
1501	63
2001	63
2501	63



O(1)

Remove From Structure

SET SIZE	AVERAGE TIME OF 1000 TRIALS
1	472667
501	45720
1001	77789
1501	100744
2001	136859
2501	217149
100000	5447043



O(N)

Since the add and remove operations are very clearly O(1) and O(N) respectively, this structure is a doubly linked list, as supported by the linear search time.