Matthew Moore

Mr. Bowers

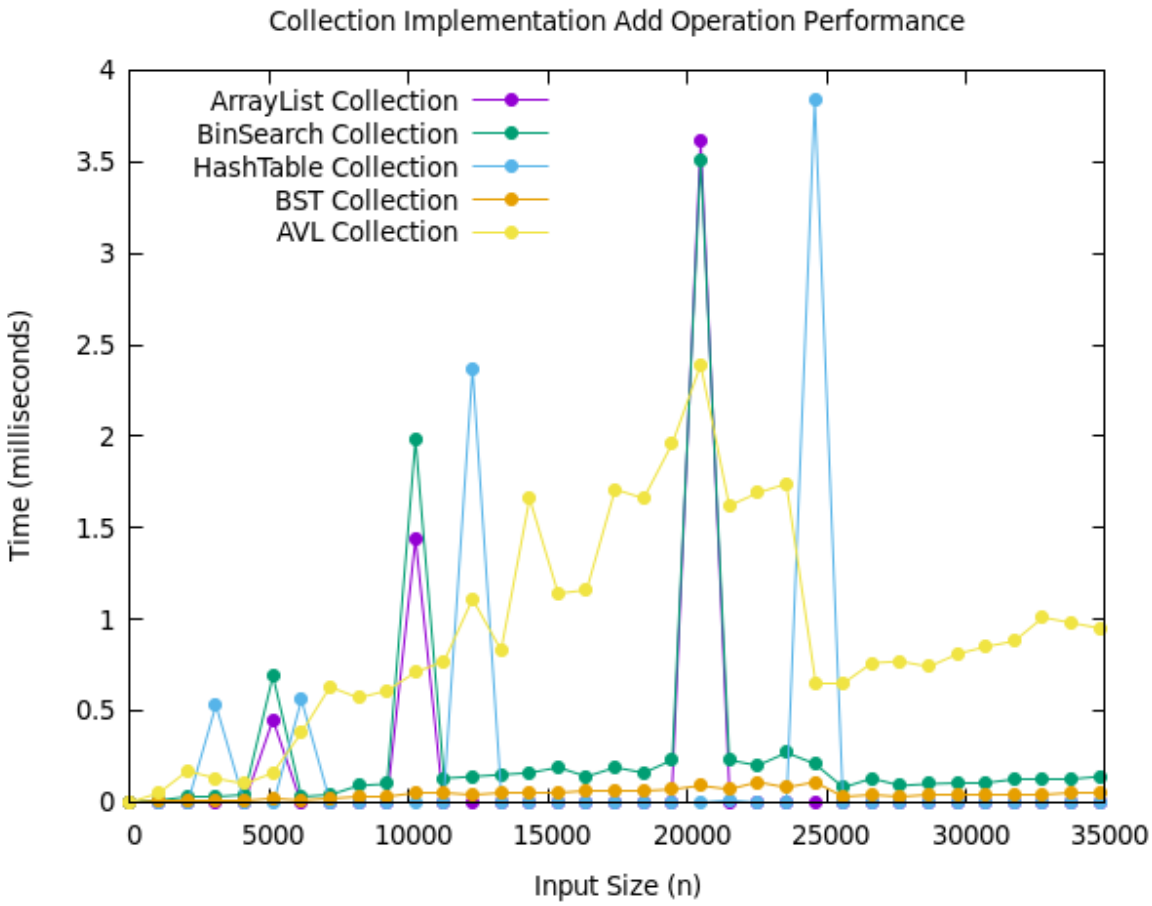
12/2/20

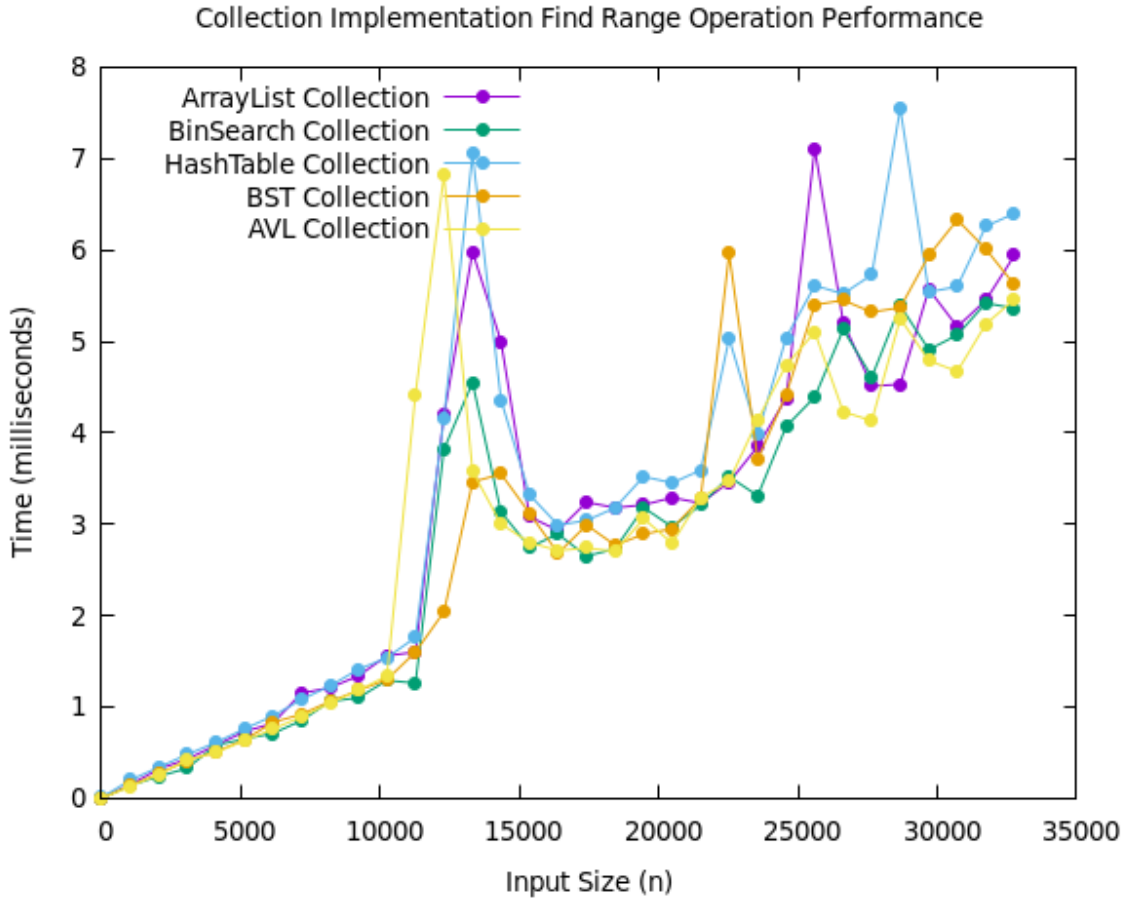
HW 8 Graph Analysis

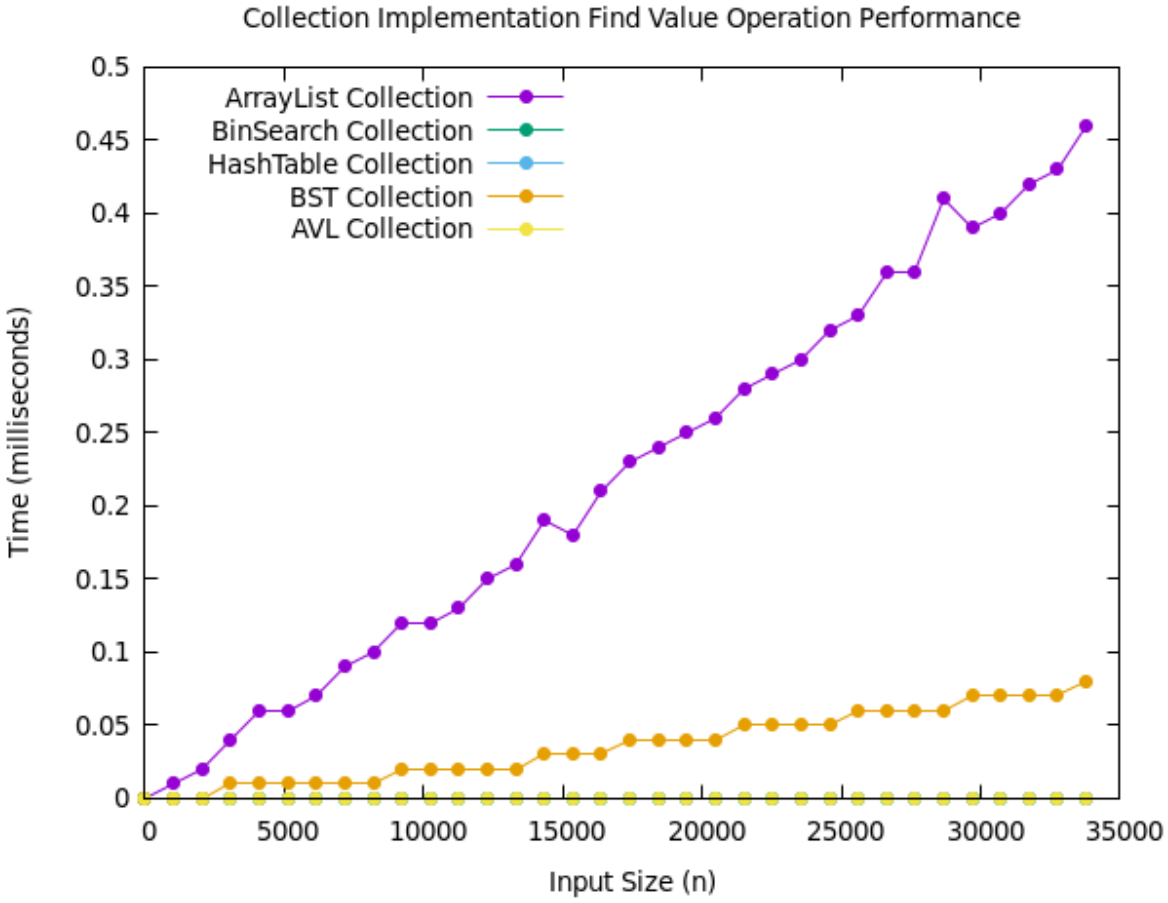
Conclusion:

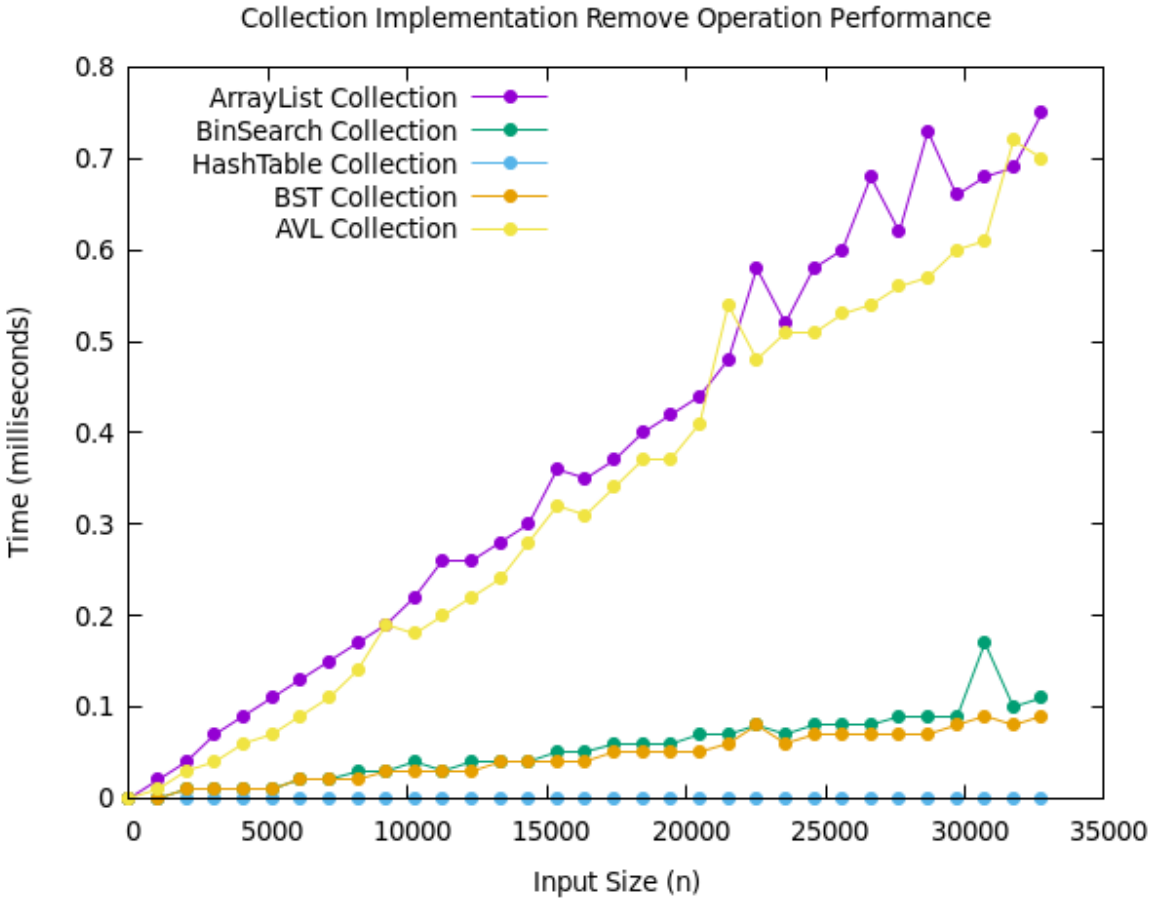
Going into the performance tests, I would expect the AVL collection to be just as efficient as the BST collection, but it seems to be much slower for the add and remove operation because of the rotations. Yet, the AVL shows to be far faster than the BST collections whenever there is any sort of search for an element. This is because the maximum height of any single path in the tree is always going to be log n. Meanwhile, the BST collection can have paths of any size so searching can get out of hand easily.

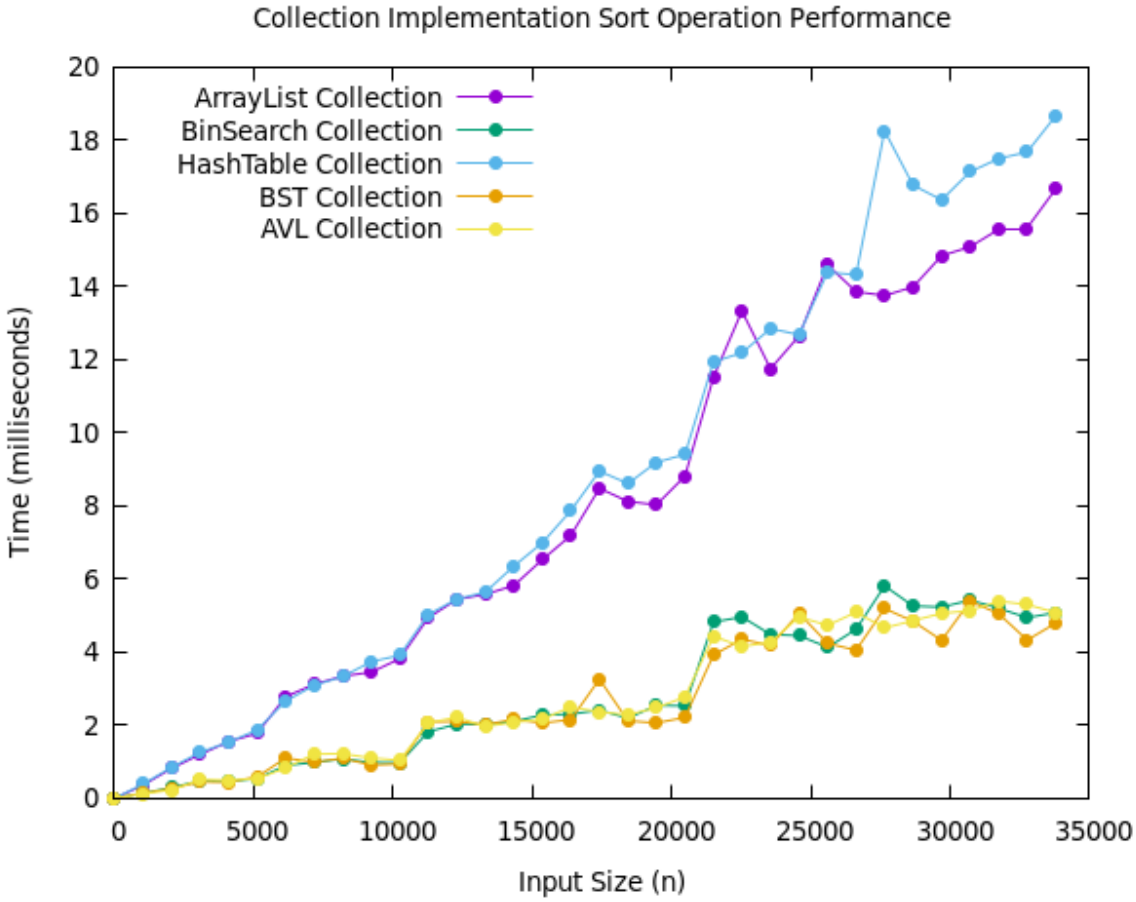
I did not find a lot of trouble with this assignment, but I struggled to figure out why my AVL collection is so slow with its add and remove operation. I think it may be due to the fact that I call the height function too many unnecessary times but I am unsure.

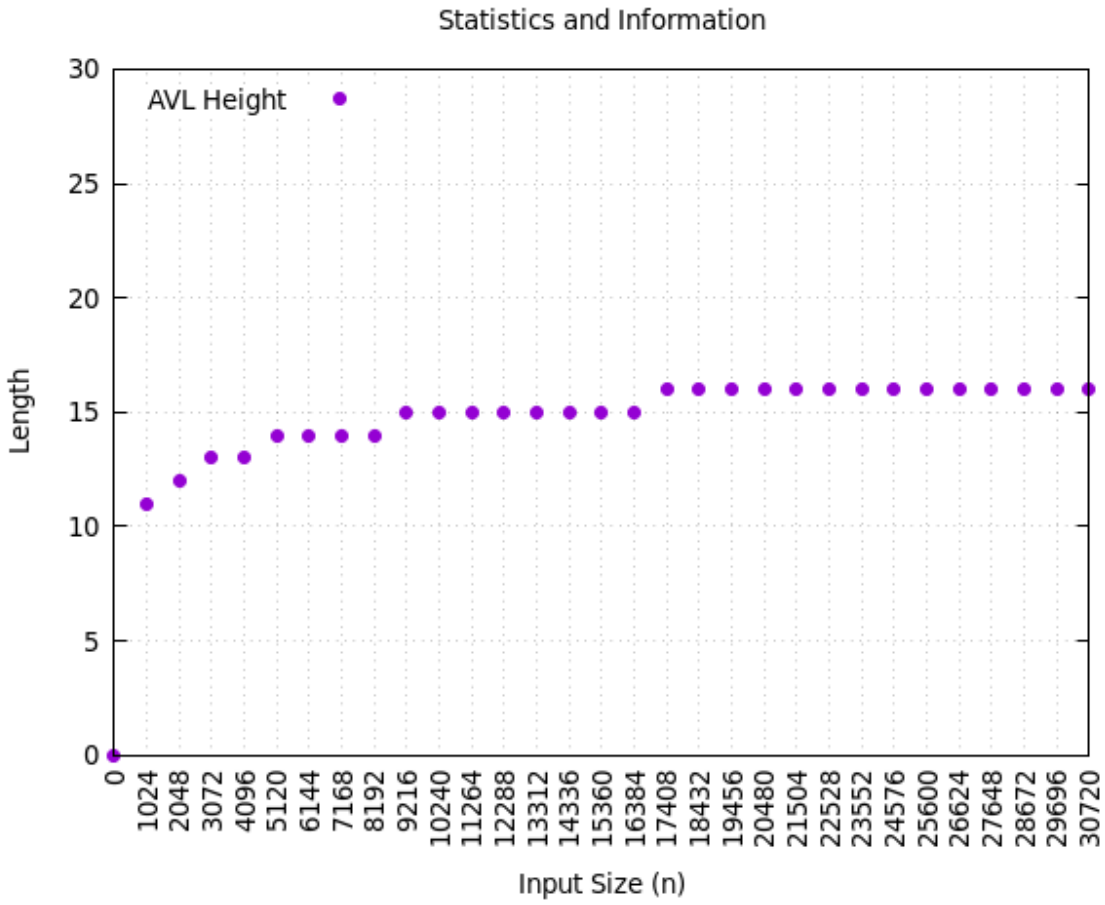



Collection Implementation Find Range Operation Performance
ArrayList Collection
BinSearch Collection
HashTable Collection
BST Collection
AVL Collection
Time (milliseconds)
Input Size (n)
0
1
2
3
4
5
6
7
8
0
5000
10000
15000
20000
25000
30000
35000


Collection Implementation Find Value Operation Performance
ArrayList Collection
BinSearch Collection
HashTable Collection
BST Collection
AVL Collection
Time (milliseconds)
Input Size (n)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0
5000
10000
15000
20000
25000
30000
35000


Collection Implementation Remove Operation Performance
ArrayList Collection
BinSearch Collection
HashTable Collection
BST Collection
AVL Collection
Time (milliseconds)
Input Size (n)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
5000
10000
15000
20000
25000
30000
35000


Collection Implementation Sort Operation Performance
ArrayList Collection
BinSearch Collection
HashTable Collection
BST Collection
AVL Collection
Time (milliseconds)
Input Size (n)
0
2
4
6
8
10
12
14
16
18
20
0
5000
10000
15000
20000
25000
30000
35000



Operation	ArrayList	LinkedList	SortedArray	HashTable	BST	AVL
Add	O(1)	O(1)	O(n)	O(1)	O(n) Avg: log n	O(log n)
Remove	O(n)	O(n)	O(n)	O(1)	O(n) Avg: log n	O(log n)
Find-value	O(n)	O(n)	O(n)	O(1)	O(n) Avg: log n	O(log n)
Find-range	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
Sort	O(n^2)	O(n^2)	O(n)		O(log n)	O(log n)