



Input	Time(ms)	BST insert	Input	Time(ms)	AVL insert
1000	0		1000	4	
5000	1		5000	13	
10000	2		10000	20	
50000	10		50000	75	
75000	12		75000	102	
100000	14		100000	125	
500000	33		500000	530	

Input	Time(ms)	BST search	Input	Time(ms)	AVL search
1000	0		1000	0	
5000	1		5000	1	
10000	2		10000	2	
50000	5		50000	3	
75000	5		75000	5	
100000	6		100000	6	
500000	22		500000	22	

Comparing the insertion time required for the AVL and BST trees, we can see the AVL tree takes significantly more time to finish inserting all of the values. This is due to the AVL tree requiring time for balancing(rotations) at each insertion/

Comparing the search time required for the AVL and BST trees, we can see that the time required to search for a set of elements is approximately the same for each tree  
I must be doing something wrong because the search time for an AVL tree is at worst  $O(\log(n))$  while the BST has a worst case time complexity of  $O(n)$ . Maybe I am just not hitting these edge cases