

## Assignment 6

Question 1:

In this question, I used 1 – 1024 to generate random numbers. And used 1025(or any number outside the range of 1 - 1024) as target to do the search function.

1	BST visited 12 nodes.	Linked-list visited 1024 nodes.
2	BST visited 10 nodes.	Linked-list visited 1024 nodes.
3	BST visited 8 nodes.	Linked-list visited 1024 nodes.
4	BST visited 9 nodes.	Linked-list visited 1024 nodes.
5	BST visited 6 nodes.	Linked-list visited 1024 nodes.
6	BST visited 9 nodes.	Linked-list visited 1024 nodes.
7	BST visited 8 nodes.	Linked-list visited 1024 nodes.
8	BST visited 6 nodes.	Linked-list visited 1024 nodes.
9	BST visited 7 nodes.	Linked-list visited 1024 nodes.
10	BST visited 5 nodes.	Linked-list visited 1024 nodes.

On average BST visited 8 nodes( $O(\log n)$ ). It is fewer than Linked-list. And if repeated more times the average visited nodes should be closed to 10. Because  $\log 1024$  is 10.

Question 2:

In this question, I used 1- 1024 to fill the BST and linked-list in order from 1 to 1024. And used 0 and 1025 as the targets to do the search function.

Table 1 used 1025 as target.

1	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
2	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
3	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
4	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
5	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
6	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
7	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
8	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
9	BST visited 1024 nodes.	Linked-list visited 1024 nodes.
10	BST visited 1024 nodes.	Linked-list visited 1024 nodes.

Table 2 used 0 as target.

1	BST visited 1 node.	Linked-list visited 1024 nodes.
2	BST visited 1 node.	Linked-list visited 1024 nodes.
3	BST visited 1 node.	Linked-list visited 1024 nodes.
4	BST visited 1 node.	Linked-list visited 1024 nodes.
5	BST visited 1 node.	Linked-list visited 1024 nodes.
6	BST visited 1 node.	Linked-list visited 1024 nodes.
7	BST visited 1 node.	Linked-list visited 1024 nodes.
8	BST visited 1 node.	Linked-list visited 1024 nodes.
9	BST visited 1 node.	Linked-list visited 1024 nodes.
10	BST visited 1 node.	Linked-list visited 1024 nodes.

From these two table, we can see BST visited different nodes when the target is 1025 and 0, respectively. As we inserted 1 – 1024 in order, the BST nodes only have right children and the left children are NULL. So, when we are searching 1025, we visited every right child of the total 1024 nodes, because 1025 is greater than any number in this tree. But, when we are searching 0, we just visited the root, because 0 is less than any number in this tree. When visited the root, then compare 1 with 0, 0 is less. So we check the left child of the root node. But the left child is NULL. Thus we just visited 1 node in this situation. Furthermore, if we insert 1 – 1024 in backorder, the result should be changed. Because the nodes in this tree only have left children, so when we are searching 1025, the number of visited nodes should be 1 and the number of visited nodes for 0 should be 1024. The results just depends on what order we chose and which number out of the range we want to search.