1. Yubin Lim, 2/28/19, CS2150-103

2.

In the cases of all three text test cases, the path taken to ‘x’ is shorter in the AVL tree than the BST. This is due to the BST allowing you to keep adding on to a node, while AVL tree makes you rebalance if there is an unbalanced tree with balance factor of -2 or 2. This makes AVL paths to find a key shorter on average.

3.

With all the rebalancing, AVL trees are able to guarantee Big theta logn running time, while the worst case for BST can be n. AVL takes up more space because you have to store the height and rebalance each time there is a balance factor of -2 or 2.

4. An AVL tree would be preferable to a BST when you have to lookup value, or paths more frequently, since the time complexity is more efficient in AVL trees. Also, if the tree is long-lived, AVL trees are better.