Writeup:

1. Fort the initial 2000000 inserts, the BST had the shortest time. Because of this the BST was the best for the initial inserts by a slight margin.
2. For the searches, the RedBlackBST had the best runtime out of all of them. This is probably because it is guaranteed to stay balanced regardless of how many nodes are in the tree.
3. The IterativeBST’s time was signigicantly shorter than the other two trees which makes it the best at deleting nodes.
4. The ordered inserts made the IterativeBST and the BST’s height to increase significantly when compared to the RedBlackBST. This is because the RedBlackBST will always remain balanced while the IterativeBST and the RedBlackBST would be skewed to the left or to the right. This would usually have a negative effect on, in the long run, get() and put() runtimes.
5. I would say that the IterativeBST performed a bit better than the BST because it had faster run times for the get() and delete() methods. This difference was especially apparent when testing the delete() methods.
6. The RedBlackBST would usually get outperformed when testing the put() and delete() methods. This is especially true when comparing the RedBlackBST to the IterativeBST since the BST would sometimes get outperformed by the RedBlackBST in some of these cases.