The way to run my code is simply compile the tree.cpp and run it. The result will be shown in the tree.out file, no extra command is needed. I have made some modifications to both the AVL tree and the Splay tree since the original answer is not necessarily shown. But still, I have passed the OJ BBST question for both of the trees. The program that I submitted to the OJ system is also included in the zip file(AVLTree.cpp and SplayTree.cpp), please ignore them when executing the project program.

In this project, I have constructed two different types of trees: the AVL tree and the Splay tree. As we have learned in the lecture, although all the required functions are O(logn), the Splay tree will splay the latest node to the top, making more switches. To Highlight this feature, I choose four different ways to generate data sets.

The first way is to pick numbers randomly so that splaying will be meaningless. The second way is to select the first value and randomly decide the next value should be the previous value +1 or -1. The next value of the third way is simply the previous value +1. The last way is to skip all the 4th functions because the value of the 4th function is irrelevant to the node value. The number of the test cases is 1 million, and the value interval is 1~1000.

Result:

generator1, AVL, time: 7.152

generator1, Splay, time: 10.569

generator2, AVL, time: 7.058

generator2, Splay, time: 7.68

generator3, AVL, time: 7.542

generator3, Splay, time: 6.75

generator4, AVL, time: 8.317

generator4, Splay, time: 6.974

The result is very close to my presumption. For generator1, All the splaying times are in vain, so the AVL tree is significantly faster. But for generator2, because the next value is very close to the previous one, the speed of the Splay tree is getting closer to the AVL tree. For generator3, the next value will continuously get farther away from the first node, the AVL tree will be inefficient, and the Splay tree will become faster. At last, for generator4, the performance of the Splay tree surpasses that of the AVL tree to a more significant extend.

After finishing this project, I learned a lot from it. Firstly, this is the largest programming project I have ever done. I discovered many debugging and acceleration techniques while doing it. Moreover, this project requires us to deal with input and output file, which I have never encountered before. The last and the most important thing I learned is that there is no ultimate solution for the data structure. Although the Splay tree seems to be the advanced version of the AVL tree, it can still be slower under many circumstances.