

In this project, you are required to write a program to compare the performance of AVL and Splay trees based on the two criteria: the total number of comparisons and the number of rotations.

You will be given a text file as input, and your program will read the characters in the text and insert the non-existing ones as keys in the corresponding tree (AVL or Splay) or otherwise (i.e., if existent) find them and update their occurrence frequency in the text. For the AVL tree, if there is an AVL condition violation after inserting the new node, you will employ the proper rotation and make the tree AVL. For the splay tree, you will make the necessary splay(s) after reading each character in the text.

The number of comparisons will be considered for both the successful and the unsuccessful searches (i.e., insertions). The number of rotations in AVL trees will be considered only when there is a need for rotation. A single rotation in AVL trees costs one time unit (*tu*) (which is equal to the cost of the comparison of two keys), while a double rotation costs two *tus*. A splay in splay trees costs as many *tus* as the number of depth levels the nodes have moved through. In Splay trees, on the other hand, there will be rotation(s) after reading each character from the text (remember Splay trees!).

The output of your program will be:

the total cost (= the cost component from the comparisons + the cost component from the rotations) of the construction and search within both trees.