Assignment-3:

Implement dictionary ADT (Abstract Data Type) using treap by taking random priority. You need to implement at least insert, delete, and search. Build an interactive user interface that performs insertion, deletion, searching and printing.

Compare the performance of treap with BST (Binary Search Tree) and AVL tree. For performance evaluation, perform several insertion and deletion in interleaved fashion. You should generate testcases into a file. Your program must read testcase file to generate several parameters such as total no of key comparison during insertion and deletion, height of the final tree, average height of each element in the final tree, total number of rotations (double rotation in AVL tree can be counted as 2 single rotations) etc.

Generate several testcase files by varying the total number of operations and compare them using graphs/charts/tables. Also compare between theoretical and empirical results. You may use additional parameters of your choice that help in result analysis.

A sample test case file is shown below:

10000 #Total number of operations

Insert 10

Insert 26

Insert 12

Delete 10

Insert 11

Delete 12

...

Points: Treap implementation – 30, Testcase generation and parameter calculation – 20, Report: Analysis of the result – 50.

Note: Use your previously written code for BST and AVL tree.

Deadline: Oct 5, 2021.