CS2002301 2021 Fall

Homework#2

Due date: 2021/11/10 08:00 (UTC+8)

1. This assignment weight 8 points.
2. Submit the code to Online judge system (https://nlp.csie.ntust.edu.tw:2021) and submit the report and source code to Moodle system. In the report, briefly explain how you solve the problem and list resources you referenced. Please followed the Homework rules and Online judge guide.
3. The source code you submit to Moodle should be the code your "last submit" to Online judge and **TAs will calculate your score by your "last upload"**.
4. If you have any question or confusion, feel free to ask on Moodle forum.

Problem 1 - 瓦基 the gardener(5 pt.)

Test case scores: 40+40+20

Thanks to your help, 瓦基 has found the shortest path out of 魯拉拉's house. However, when 瓦基 tries to go through 魯拉拉's garden, he is hindered by the overgrown Binary Search Trees.瓦基 must prune the BSTs ASAP so as to escape from 魯拉拉's house keepers. There are 5 operations that 瓦基 can do:

1. Insertion: 瓦基 inserts a number, *I*, to the BST. Smaller numbers will be on the left. If there are duplicated numbers, insert the latter one to the left subtree of the former one.
2. Deletion: 瓦基 deletes a number, *D*, from the BST. If *D* has both left and right child nodes, find the greatest number in the left subtree to replace *D*. If *D* has only one child node, link the child node to its parent node. If *D* does not have any child node, simply eliminate *D* . If duplicated *D*s are found in the tree, eliminate the one nearest to the root.
3. Print: 瓦基 traverses the BST in preorder. The traversal should be noted in format root(left__child_node()())(right_child_node()()).

4. remove value with Least node: 瓦基 wants to make the total value of the BST at most *V*, please return the least number of nodes he needs to remove.
5. remove value with Most node: 瓦基 wants to make the total value of the BST at least *V*, please return the most number of nodes he needs to remove.

Please help 瓦基 estimate the result of certain commands.

Constraints:

- Do not use any libraries (vector, queue, stack…) except for standard I/O. Please implement your own class using array if you needed.
- TAs will manually check your code after the deadline.

Problem 2 - Students' score manage system(3 pt.)

Test case scores: 34+33+33

The TAs have finally done correcting all the midterm exam papers, only one problem remains. Since Data Structure is such an intriguing subject that everyone wants to study, there are a great deal of students taking this course. Managing the students' score thus becomes a burden. Fortunately, Hank designs a system to manage the scores with AVL tree. The system includes the following functions.

1. Insertion: insert a number, *I*, to the AVL tree. Smaller numbers will be on the left. If there are duplicated numbers, insert the latter one to the left subtree of the former one.
2. Deletion: delete a number, *D*, from the AVL tree. If *D* has both left and right child nodes, find the greatest number in the left subtree to replace *D*. If *D* has only one child node, link the child node to its parent node. If *D* does not have any child node, simply eliminate *D*. If duplicated *D*s are found in the tree, eliminate the one nearest to the root.
3. Print in preorder: traverse the AVL tree in preorder. The traversal should be noted in format root(left__child_node()())(right_child_node()()).
4. Print in inorder: traverse the AVL tree in inorder. The traversal should be noted in format (()left__child_node())root(()right_child_node()).

5. Print in postorder: traverse the AVL tree in postorder. The traversal should be noted in format (()()left__child_node)(()()right_child_node)root.

Please help Hank implement the system.

Constraints:

- Do not use any libraries (vector, queue, stack…) except for standard I/O. Please implement your own class using array if you needed.
- TAs will manually check your code after the deadline.