PROJECT2

Chinese-English Dictionary based on Binary Search Tree

I. Summary

To better understand the tree data structures learned in this course, the project designed for you to use AVL tree and red-black tree to implement an English-Chinese dictionary.

II. Percentage Point

Red-Black tree(30%):

initialization(8%): Initialize the tree with init.txt
Delete(6%): Delete the words with delete.txt
Insert (5%): Insert new words with insert.txt
Search (6%): All correct search
Preorder print (5%): Print the tree in right format to rbt.txt

AVL tree(30%):

initialization(8%): Initialize the tree with init.txt
Delete(6%): Delete the words with delete.txt
Insert (5%): Insert new words with insert.txt
Search (6%): All correct search
Preorder print (5%): Print the tree in right format to avlt.txt

General(10%): Correct implementation
Document(20%): Full and detail. Reasonable analysis
Coding style(5%): Proper comment
Bonus(5%): Creative thought

III.Requirements

A. Functions

Implement red-black tree and AVL tree by yourself. The core functionalities must be implemented in C++ from scratch(including **INSERT** 、**DELETE** 、**SEARCH**、**PREORDER_PRINT** methods) .

a) PREORDER_PRINT

Display the tree by PREORDER traversal to a file.

Here we give an example after insert into trees with number 7, 3, 5, 1, 6. We require you to print like that format. In this project, you are supposed to replace the number with words.

Print result for red-black tree

```
level=0 child=0 5(BLACK)
level=1 child=0 3(BLACK)
level=2 child=0 1(RED)
level=3 child=0 null
level=3 child=1 null
level=2 child=1 null
level=1 child=1 7(BLACK)
level=2 child=0 6(RED)
level=3 child=0 null
level=3 child=1 null
level=2 child=1 null
```

b) INSERT/DELETE

1. A batch of words which contain in a file. *The first line in the file represents which operation it will do and next lines are data.* Each time you operate with one file, please call your PREORDER_PRINT method in tree classes to print the tree to rbt.txt / avlt.txt.

2. A single word.

c) SEARCH

1. Some words in range and give their meanings. E.g. we give a query: please search from 'aa' to 'apc', then you give the words between 'aa' and 'apc' as well as their meanings. The boundary values don't have to be exactly words.

2. A single word. Just give its Chinese meanings.

Hint: You know that English words are in lexicographic order. So please use the word as key and build the trees.

B. Analysis work

Another work you should do in this project is to compare the operations like insertion and deletion for both trees.

Please call your methods in the following ways for each kind of tree, and analyses each time they spent.

Note: Don't change the files we have provided or disarrange the step order.

1. insert into trees the data in the file init.txt
2. delete the data in the file delete.txt
3. Add the data in the file insert.txt
4. Query a word
5. Query some words

For the first three steps, after each operation on 100 pieces of data, you should record the time used. And finally give a document based on this analysis and testing time should be attached on it.