

Project 2

English-Chinese Dictionary based on Binary Search Tree

1. Summary

To better understand the tree data structures learned in this course, the project designed for you to use red-black tree and B+ tree to implement an English-Chinese dictionary. In case that you have not learned much about B+ tree on class, we provide you a learning material.

2. Requirements

a) Functions

Implement red- black tree and B+ tree with branch number 50 by yourself.

(including **INSERT** 、 **DELETE** 、 **SEARCH**、 **PREORDER_PRINT** methods)

PREORDER_PRINT

Display the tree by PREORDER traversal on console.

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
```

Print result for B+ tree

(Here we show the tree with branch number 4, but you are required to use branch number 50 in this project)

```
level=0 child=0 /5/
level=1 child=0 /1/3/
level=1 child=1 /5/6/7/
```

Then use two trees to implement an English-Chinese dictionary for each. Implement **INSERT** , **DELETE** , **SEARCH** methods for the dictionary.

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. To learn more about the file format, you can see the sample files we provide.

Each time you operate on 100 pieces of data, please call your **PREORDER_PRINT** method in tree classes to print the tree on the console if the total data size in tree is not larger than 500.

2. A single word

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.

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

1. First insert into trees the data in the file 1_initial.txt
2. Then delete the data in the file 2_delete.txt
3. Add the data in the file 3_insert.txt
4. Query a word
5. Query some words

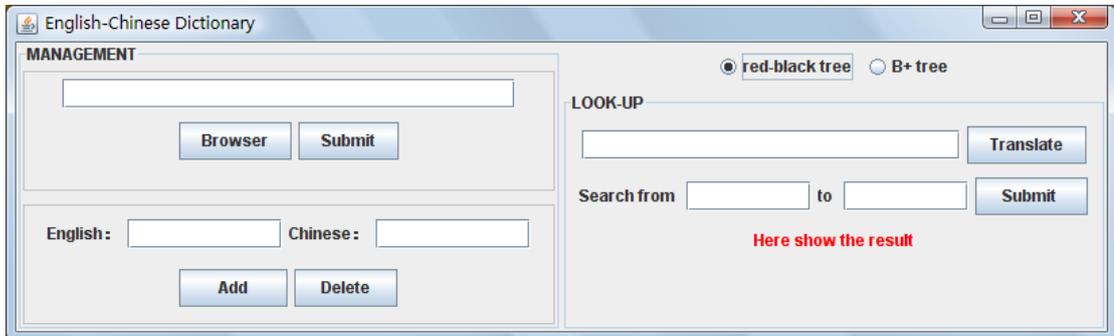
For the first three steps, after each operation on 100 pieces of data, you should record the time used. For step 4 and step 5, just give the time totally consumed for each query.

And finally give a document based on this analysis and testing time should be attached on it.

3. Design

To make your program more flexible, we don't recommend that you input the command files by hard-coding. We prefer that you provide an interface through which the user can import their files.

Here we provide a demo.



Explanation:

The left is management part.

This part gives two ways to extend or modify the dictionary.

1. The user can choose a file and execute the corresponding operations.
2. The user can add or delete a single word.

If you just need to delete a word, only fill the English word.

The right top is choice part.

It means which kind of tree you are using to implement the dictionary.

The user can make the choice.

The right down is look-up part.

In this part, you are required to search words in the dictionary.

1. Just give a translation for the word in the form.
2. Or give the words and their translation results in the query range.

The upper is just a demo. We will be glad if you can show us a more user-friendly interface.

Remind you that you should also print the tree forms on the console. See the `PREORDER_PRINT` part in functions requirements for detail.

4. Percentage points

Items	Cost	Description
General	65%	Correct Implementation
UI	10%	User-friendly and easy to use
Documents	20%	Full and detail. Reasonable analysis
Coding Style	5%	Proper comment
Bonus	5%	Creative thought.

5. Hand in requirements

a) Deadline

2011-11-06 Sunday 23:59:59

Please submit to

ftp://10.132.141.33:21 /classes/10/111 数据结构与算法设计(郑晓庆)
/WORK_UPLOAD/Project1/

b) Documents

1. All the codes
2. A document based on the analysis

Suggestion: Start early and before you begin to code, please fully understand B+ tree and red-black tree.

If you have any questions, you can contact us. Good luck!