Design Document

Brian Pina-Estrada

1.AddressBook Program, Brian Pina, 05/09/2016

Report::My program reads in and does most of the tasks, but I couldn't implement it to work with the tree with the sort and search functions, I'm also getting an error with my affiliates they seem to have been given id values for the First name on some and others repeat when I print them out, my search book isn't very accurate since you have to input the whole field correctly, update works fine and so does print, change the the input using dos2unix helped on a issue with the id being in the first name field. I also created and changed the bstree to avoid some errors with overloading and to print out the tree and write out the tree.

2. Problem Description:

This Program will manage a Addressbook of a certain amount of people from a input file, the user will have the ability to add, delete, update an entry, as well as sort by a certain field. The user will also be able to search the book from any given field and print out the whole book, at the end the user will able to select what fields they want printed out to a file as well as whether they want to print it out to a new file or the field they read from.

There is a read file function which will be used to read in the file and fill in the tree and a vector that will be used to sort. The main menu function will return an int based on their choice and that choice will call a certain function to manage the address book,

Sourcebook will ask the user what they want the field to sort by then the program will output the sorted output.

Search Box will ask the user what field they want search by annd then will be asked to enter what they want to search by, if the field is found the program will output the whole field.

Update Book will have three options where the user will be asked if they want to update ,delete or add an entry, within update the user will be shown the address book and will chose what they want to update followed by what field they want to update, delete will show the affiliates and then will asked which one they will delete. Add will add a whole entry the user will be asked to add each field plus the amount of affiliates and fill their fields.

Print will call a modified print function that will show the entire avl tree to the user.

4. Input Requirements

The data will be retrieved from a stream file and will be read into the program and fill the tree, another file will be used to print out the data or the same will be used depending on the user's input with their choice.

Most of the menu items and choices will be of int values where they are given a predefined amount of options if an option is not inputted they will be asked to input again this also for the case if they input the wrong data type like a char instead of a int.

When searching and sorting items the user is allowed to enter string values for each input used by a getline function

5. Output Requirements

The output will come from the print list and other print functions that will either show the entire tree or just a subject the valid forms are only the whole tree itself or a single subject.for Output to a file the data will newline delimited for each field except for affiliates where a whole line is affiliates separated by commas, while a whole entry is divided by ‘|’.

6. Problem Solution Discussion A summary description of the solution steps with algorithms analysis (1paragraph, approximately 100 words). If any unusual techniques or algorithms are used that need further explanation, and additional paragraph may be used.

My uncommon algorithms techniques are the use of vectors for sort and search since I wasn't able to implement them with the bstree.cpp another one is the use of an array for the denomination change to avoid a bunch of if statements and cin this used for the writebook to determine what fields will be written out for the user, another practice is the use of print and write functions that derive of the original bstree print functions these were used to show one whole node while it used recursion to traverse the tree and the same with write except it prints to the output file.

7. Data Structures

A description of choice of your data structures and justification. Of course the main data structure for the database is AVL tree. But for some internal operations you may have to use a few sub containers. So include a brief explanation for yourchoice. For example, "I have considered DS1, DS2, and DS3. Their pros and cons are summarized as follow... I choose DS1 over DS2 and DS3 because ...."

I mainly used the avl tree as the main structure to hold the address book but I also used a Address Book container to be used for sorting from a certain field and searching since I couldn't figure out how to implement that with the tree, I also used a denominations array to hold the denominations for what field should be printed out so that functions just used the array to select what fields will be printed out

8. User Interface Scheme

User interface scheme should show the menu items at top level and items in submenus and how to navigate through menus.

Mainmenu

1sort

2search

3update

4imprint

99 exit and save

Sort

Asks what field they want to sort by

1:First Name

2:Lastname

Etc…..

Search

Asks what field to search by

1:first name

2:lastname

Etc….

Update

1:add

2:delete

3:update

Add::will ask for input to add to the tree for one subject

Delete::

Show book

Asks which one they want to delete

Choice

Deltes

Update:shows the book

Enter the id they want to update

Which field to update

1;First Name

2:Lastname

Etc….

Print

Will ask which tree they want to show the initial one or the one that's updated if they changed something

Exit:

Will ask what file they want to save to

1 = new file

0 = same file