

ECE 250 – Project 2
Design Document
Chen-Yi(William), Liu, UW UserID:cy6liu

Overview of Classes

Class: node

Description: This class contains a pointer that points to the pointer children(array) of the node, a boolean isWord to mark whether it is a word, and char c to store letter in the node.

Member Variables:

- c: stores the letter of the word passed into the insert function.
- children: an array that represents the spot of the 26 English alphabet
- isWord: a boolean to mark whether it is a word.

Friend class trie

- so I can have access to the private and protected members of the class in Node in trie class

Class: trie

Description: The trie class links the nodes together as a trie data structure with the children pointer. This class provides operations like insert, erase, search, print, autocomplete, empty, clear, size, and output the corresponding value to the console when an action is performed.

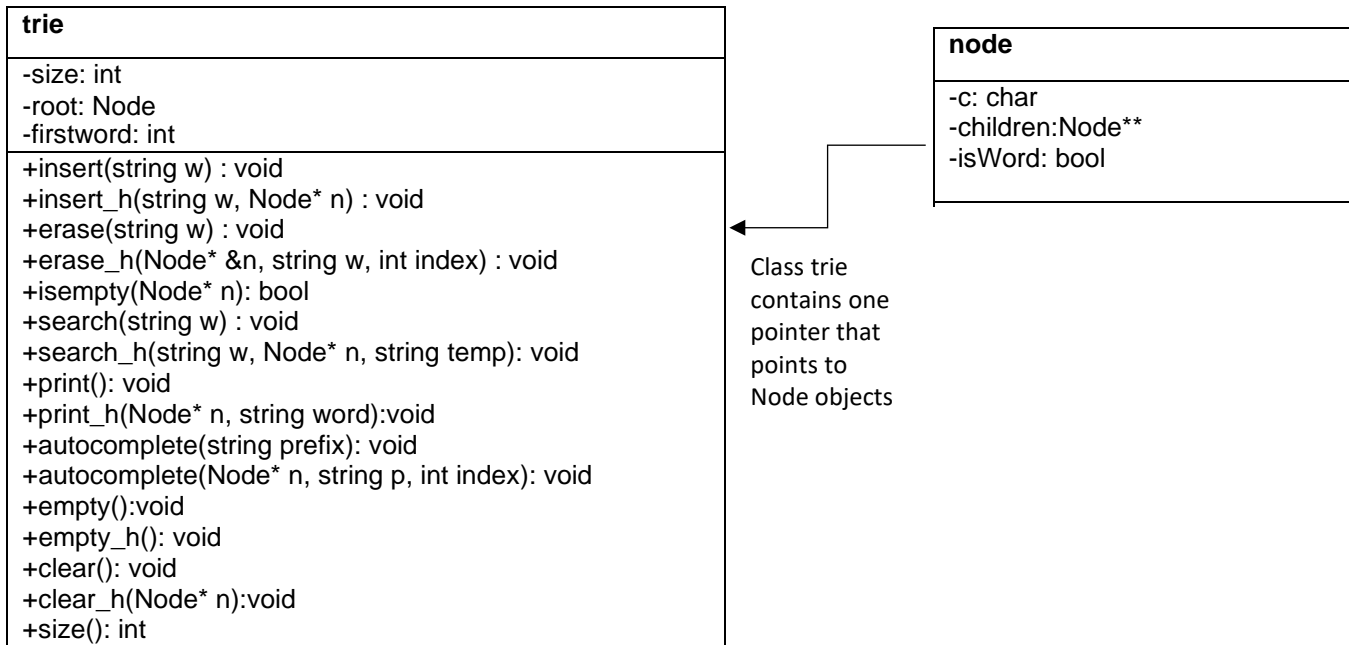
Member Variables:

- size: the current number of words in the trie
- root: a pointer the first node of the trie.
- firstword: an integer for printing a line after calling the autocomplete function, 1 to print the line.

Member Functions:

- insert: takes in a string as parameter and insert word in the trie calling the insert.h function.
- Insert_h: takes in a string and the current node to insert the word into the trie using recursion.
- erase: takes in a string as parameter and erase a word in the trie calling the erase.h function.
- erase_h: takes in a string, pointer to the current node, an index to erase the word using recursion.
- lsempy: a helper function to check whether a node is empty.
- Search: takes in a string and call for search_h to search for the word in trie.
- Search_h: takes in string word, current node, and another string to search a word using recursion.
- print: calling the print_h to print out all the words in the trie.
- print_h: using recursion to print out all the words in trie.
- autocomplete: takes in a string prefix, calling autocomplete_h to print out all the words that have the same prefix in trie.
- autocomplete_h: takes in a node, string , and an index to print out all the words of the prefix using recursion.
- empty: checks if the trie is empty and print "empty 1" if the trie is empty, otherwise print "empty 0"
- empty_h: check if the trie is empty and return true if the trie is empty, otherwise return false.
- clear: delete all the words from the trie calling clear_h.
- clear_h: takes in a current node and delete all the words from the trie using recursion.
- size: output the current number of words of the trie.

2. UML class diagram



3. Details on design decisions

- trie uses c++ default destructor. In its constructor, int size is set to 0 and root is set to null character.
- Node class uses c++ default destructor. The constructor takes in one parameter which is the letter set the private member char c to the parameter. It also sets the isWord equal to false and each spot in the children array nullptr.
- No overloading operator is used in this project

4. Test Cases

Tested 13 test cases from github

Some testing strategy I used is to test the edge cases, for example, word case-insensitivity, and executing the commands on the word in different formats and some edge cases.

Example of a test case I made myself:

```
Insert a
Insert ab
Insert abc
print
erase ab
s ab
e a
s abc
size
```

5. Performance considerations

Within the trie class, except for the empty and size functions that are $O(1)$ since they don't depend on the input values. Functions such as insert, erase, and autocomplete all operate in $O(n)$ time since those functions depend on the input values and I have achieved better run time using recursion than using for/while loop since each function call is its own closure, as a result, I don't need to take care of storing previous data, setting correct data for next loop etc.