vector<string> readWordsFromFile(string fileName)

Helper method to read lists of words from txt file for easy testing

TEST(TrieConstruction, emptyTrieNode)

Tests the construction of a Trie. After the trie is constructed, there should be 1 node in existance

TEST(TrieIsWord,emptyStringTrie)

Tests that the empty string is a word if the trie if its added to the file

TEST(TrieAdd, nodeConstruction)

Tests that the right number of nodes are added when two words with the same prefix are added to a trie

TEST(TrieDestruction, nodeRemoval)

Checks that equal number of nodes are created and destructed when an empty Trie moves out of scope

TEST(MemoryTest,newDeleteNum)

Explicited count the number of adds and deletes to ensure that there are no memory leaks after content is added to the trie

TEST(TrieDestruction, nestedNodeRemoval)

Checks that the destructor deletes all nodes that should be in a tree

TEST(AddwordTest,singleLetter)

White box test that checks that all alphabets are added to the right place in the Node's array

TEST(AddwordTest,multiwordBranch)

White box test that confirms the location of each node added to the trie by iterating through the tree

TEST(AddwordTest,addEmptyString)

White box test that checks that the node array is not affected when adding an empty string

TEST(isWordTest, emptyStringTest)

Tests that an empty trie does not contains words

TEST(isWordTest, isWordTest1)

Stress test that addword and isWord works by adding and looking for all words in a dictionary

TEST(isWordTest, isWordTest2)

Checks that is word does not return true for prefix that has not being added to the trie

TEST(allWordWithPrefixTest, test1)

Test the functionality of allWordWithPrefix by checking if it returns all words in the trie when an empty string is passed into the method. It also tests if the method returns the right list the prefix if the prefix is a word and when it isn't.

TEST(copyConstructorTest, content)

Tests the copy constructor by constructing a tree, copying it and checking if the content has being copied with isWord

TEST(copyConstructorTest, singleNode)

Tests if single node is copied by the copy constructor. Checks that a new Trie is created

TEST(copyConstructorTest, singleNodeRoot)

Test if a single node is copied by checking that a new root is created

TEST(copyConstructorTest, singleNodeRootContent)

Test if a single node is correctly copied by verifying the content of its root

TEST(assignmentOperatorTest,content)

Checks if the assignment operator is working by constructing 2 trees, adding content to the first, and assigning it to the 2nd. Checks if the 2nd trie has the content of the frist.

TEST(addWordTest,duplicates)

Tests that duplicated words cannot be added to the trie.