**BLG 381E**

Advanced Data Structure

2011 Summer School

**Report of Homework 3**

Date of Submission : 10.08.2011

Student Name : Ozan Arkan Can

Student Number : 040090573

Instructor : Arzucan Özgür Türkmen

#### Teaching assistant: Ahmet Aycan Atak

CRN: 30621

INDEX

[INTRODUCTION 3](#_Toc300695725)

[Development and Operating Environments 3](#_Toc300695726)

[Ms Windows 3](#_Toc300695727)

[Unix 3](#_Toc300695728)

[UML Class Diagram 4](#_Toc300695729)

[BUILDING B-TREE 4](#_Toc300695730)

[Getting An Article 5](#_Toc300695731)

[Splitting Article Into Sentences 5](#_Toc300695732)

[Getting Words 7](#_Toc300695733)

[Searching And Inserting 8](#_Toc300695734)

[SEARCHING KEYS 8](#_Toc300695735)

[CONCLUSION 11](#_Toc300695736)

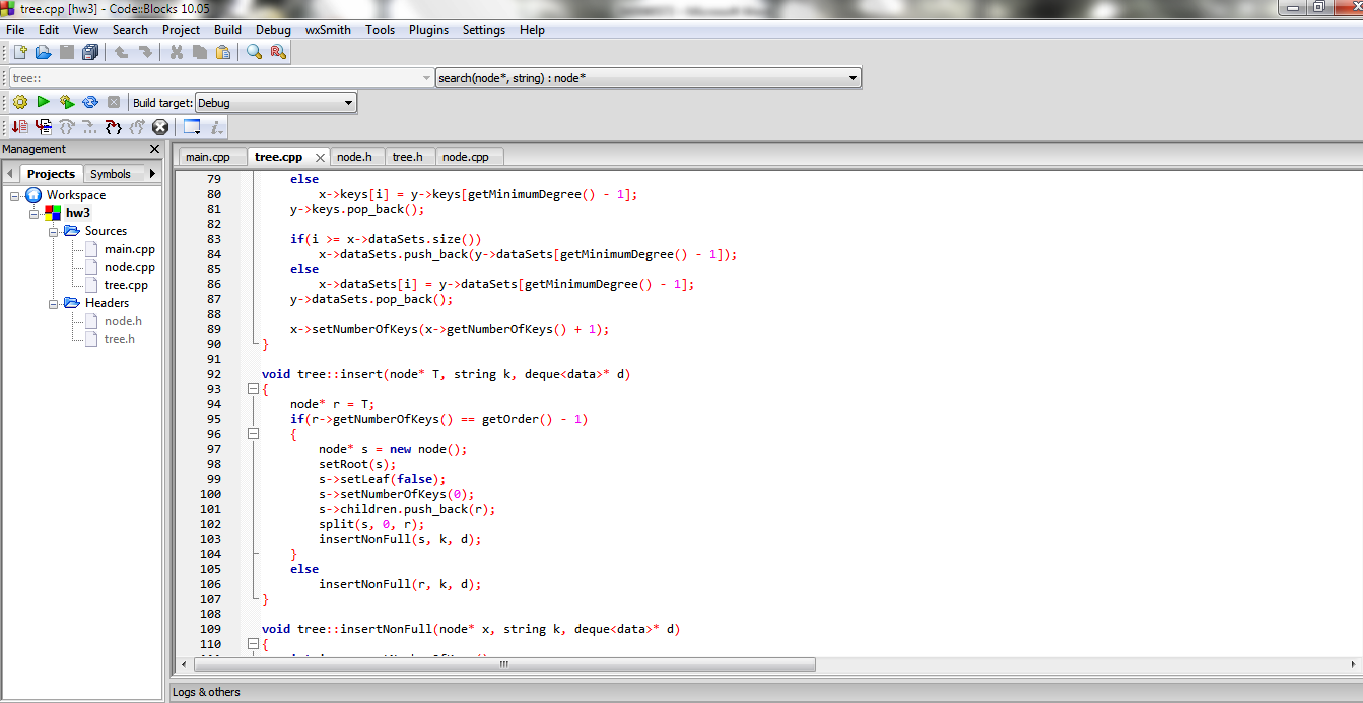
# INTRODUCTION

In this homework, it is aim that showing difference between execution time with B-tree data structure with different order values (m) for searching a key. B-tree is build for m values which is obtained from command prompt. B-tree is built using Reuters dataset. Words that obtained from Reuters dataset, is used as keys for b-tree implementation. Node n has keys and dataset that has x value, word in which article and y value word in which sentence article x, for every key. After building b-tree is completed searching in b-tree n keys which is obtained from command prompt is executed. Finally, program outputs the total execution time for searching work.

# Development and Operating Environments

## Ms Windows

The Code::Blocks IDE has been used to write source code, compile and run the code.



## Unix

The source code has been also copied to Unix, then compiled and tested with the GNU C++ Compiler. The following is the commands used:

To compile :

g++ main.cpp tree.cpp tree.h node.cpp node.h –o hw3

To Run: ./hw3 4

# UML Class Diagram

Program UML Diagram is like following chart.



# BUILDING B-TREE

Building b-tree has following steps:

-Get an article from Reuters dataset

-Split this article into sentences

-Split sentence(that is obtained previous step) into words

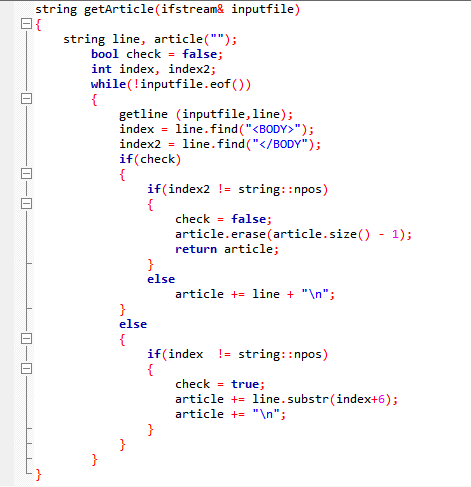
-Search that word is it in tree

True: Add data in dataset

False: Add key in tree

## Getting An Article

Reuters dataset contains article between <BODY> and </BODY> tags. getArticle function does this work. Also it moves pointer that points place in a file to forward.



## Splitting Article Into Sentences

In this work articles are represented as string. In this situation sentences are substring of article. Founding end of sentence is main idea for getting a sentence from article. Following situations are assumed as end of sentence:

-If a character is ‘!’ or ‘?’

-If a character is ‘.’

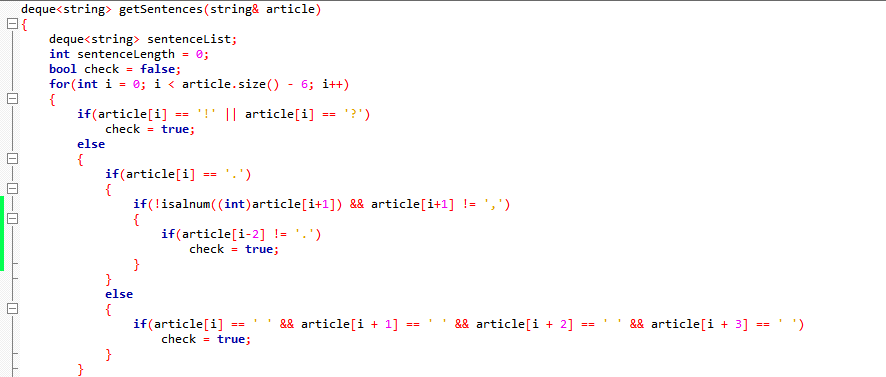
If character after that is not alphanumeric and ‘,’

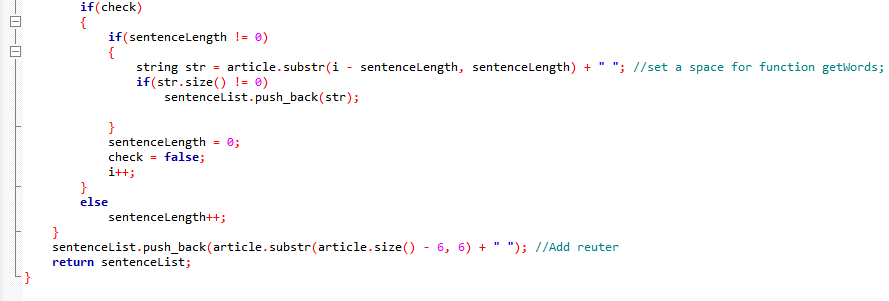
If previous 2 character that is not ‘.’

-Sequential 4 characters are space

This method is enough for much sentence, but if sentence has table it causes getting wrong sentences.

getSentences function get a string and returns list of sentences.





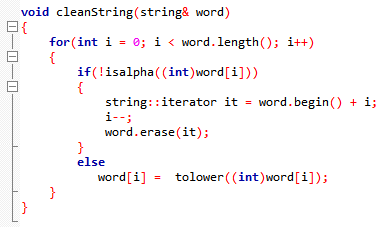
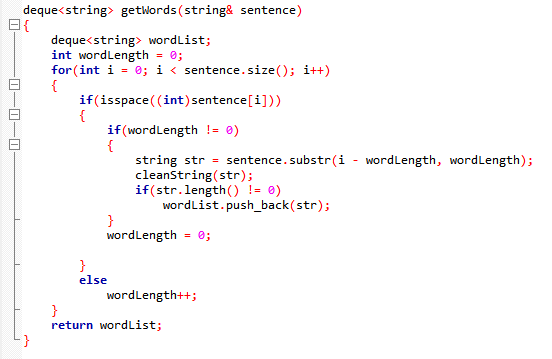
## Getting Words

Sentence splits into words from spaces. After obtaining a word following operations is applied on it.

-All characters are made lowercase

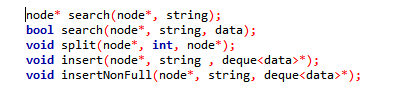
-All characters are eliminated except letters

In program, getWords function and cleanString function are used for these jobs.



## Searching And Inserting

For search and insert methods algorithms that are learned in lecture, are used.

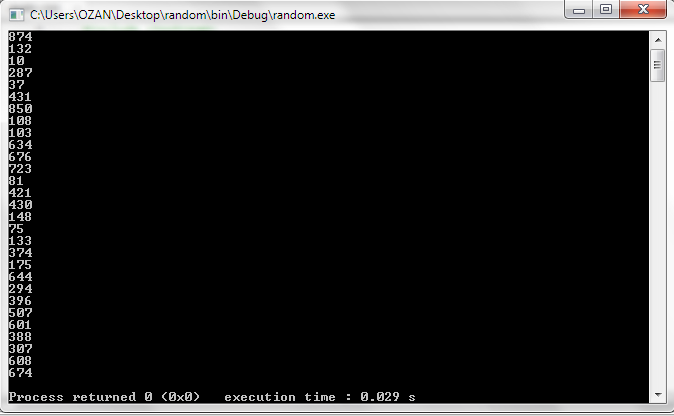


# SEARCHING KEYS

After building b-tree with m order value, it is expected that searching n keys that are obtained from command prompt in tree. If key is in tree, program outputs its data, otherwise program should warn about this issue.

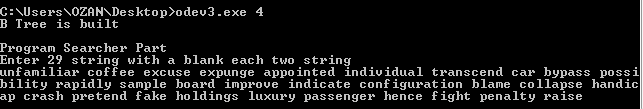
Keys that will be searched must be chosen randomly. Choosing keys randomly this method is followed.

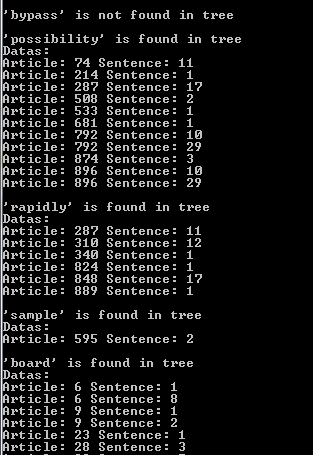
Choosing a page from Oxford Thesaurus Dictionary and pick a word from that page.



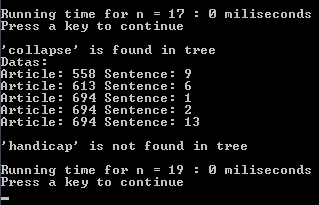
When this method is applied following words are obtained.

“unfamiliar”, “coffee”, “excuse”, “expunge”, “appointed”, “individual”, “transcend”, “car”, “bypass”, “possibility”, “rapidly”, “sample”, “board”, “improve”, “indicate”, “configuration”, “blame”, “collapse”, “handicap”, “crash”, “pretend”, “fake”, “holdings”, “luxury”, “passenger”, “hence”, “fight”, “penalty”, “raise”





Program could not calculate the total time for searching n keys in b-tree, because it is time is so little.



In this program nodes are not written in a file. And all works are done on memory. Searching a key is O(logn) for little m, because b-tree is more like binary tree. When m value is greater, execution time closes O(n) . Height of tree comes low and number of nodes is very small. This situation keys in these nodes with same layer. When execution time is 100miliseconds for m = 4 and n = 17, following histogram can be drawn.

# 

# CONCLUSION

In this homework,

* properties of b-tree
* implementing a b-tree with augmented features
* Searching a key with different m values
* Text processing

are learned.