Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 06 REPORT

Muhammed ÖZKAN
151044084

Course Assistant: Ayşe ŞERBETÇİ TURAN

1 INTRODUCTION

1.1 Problem Definition

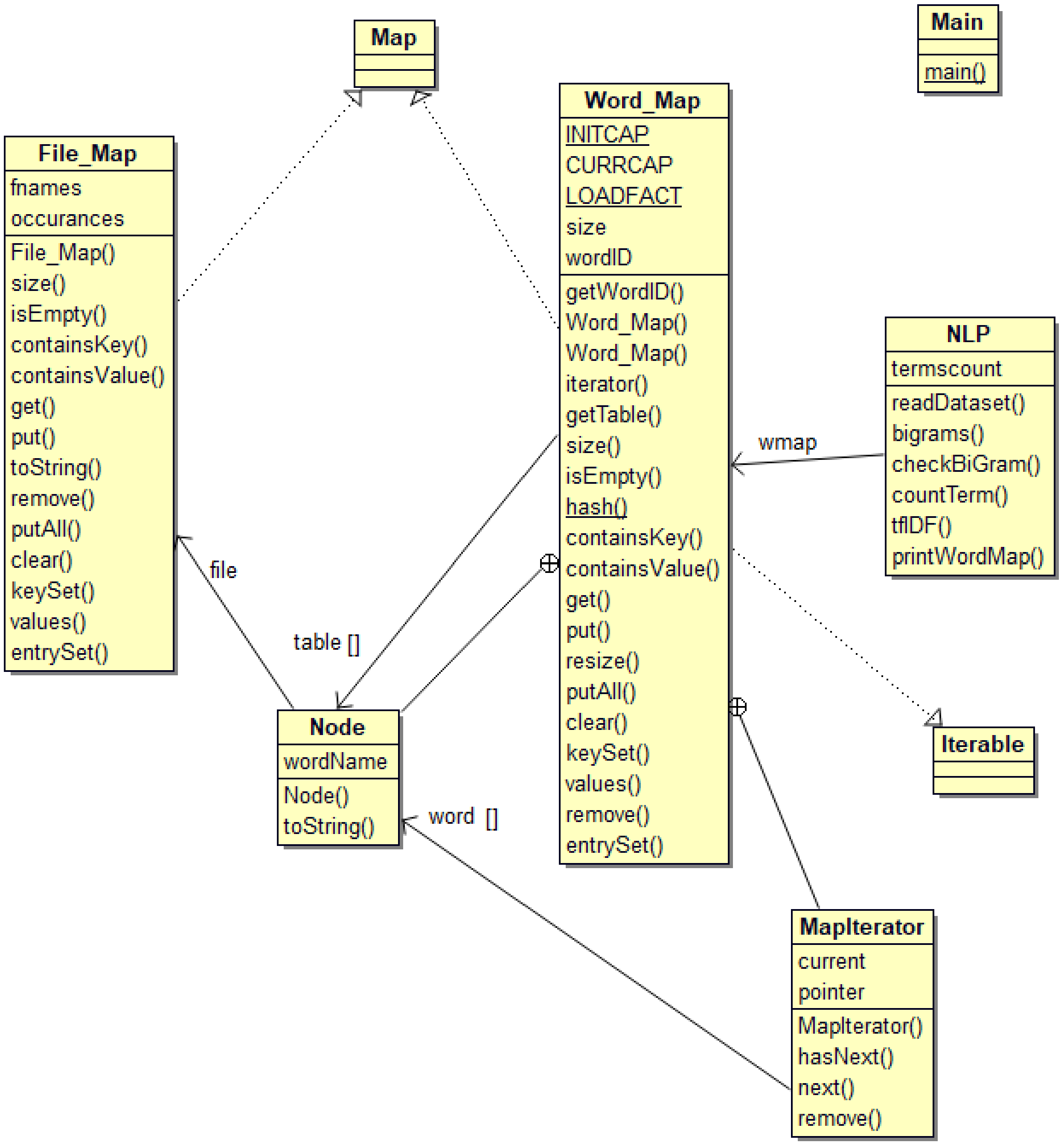
In this assignment,We will develop HashMap structure to perform Basic Natural Language Processing operations.We will read a text dataset folder consisting of multiple input files and keep the words and file name in the Word and File HashMap. The key for the word hashmap will be the words and the value will refer to file hashmap structure which keeps the occurrences of the word in different files. The key for the file hashmap is the filename and the value is an Arraylist structure containing the word positions in that file. After obtaining this structure, we will implement two basic operations used in NLP : retrieving bi-grams and calculating TFIDF values.

1.2 System Requirements

We will develop this program for all devices running Java. The classes used in the solution of the problems have been developed considering the minimum possible memory consumption and execution time. The size of the File_Map and Word_Map class on memory is to vary depending on the type of data to keep. Memory size shows a linear increase. In case of proper use, the prepared programs can be used in any environment, even on a smartphone or even on a smart watch.

2 METHOD

2.1 Class Diagrams



2.2 Use Case

The software works on the console screen. The user must specify the input file to be used as the parameter to the program before running the programs.

For example:
-java program_name input.txt

2.3 Problem Solution Approach

Since we need to develop two different HashMap structures, we will use linear probing in the first HashMap structure in the WordMap class. In the second HashMap structure, in FileMap we will use the Arraylist structure. We will implement the Map interface in both HashMap structures. We'll keep the FileMap reference in the value section of the WordMap structure. We will establish the required structure by using the relevant functions of these two classes. Then, using this data structure, we will make calculations according to TFIDF formula given to us in PDF and we will find the bi-gram.

2.4 Complexity of Functions

The complexity of functions is calculated according to the number and structure of the loops they contain. Since the complexity calculations are considered infinite, the comparison, assignment and similar operations within the functions are not included in the calculations since they do not have any meaning in infinity.

Function Name	**Complexity**	**Big O Notation**
NLP.bigrams()	$T_1(n)=n*n$	*$O(n^2)$*
NLP.checkBiGram()	$T_2(n)=n*n$	*$O(n^2)$*
NLP.countTerm()	$T_3(n)=n$	*$O(n)$*
NLP.tfIDF	$T_4(n)=1$	*$O(1)$*
NLP.printWordMap()	$T_5(n)=n$	*$O(n)$*
Word_Map.size()	$T_6(n)=1$	*$O(1)$*

Word_Map.isEmpty()	$T_7(n)=1$	$O(1)$
Word_Map.hash()	$T_8(n)=1$	$O(1)$
Word_Map.containsKey()	$T_9(n)=1$	$O(1)$
Word_Map.containsValue()	$T_{10}(n)=n$	$O(n)$
Word_Map.get()	$T_{11}(n)=1$	$O(1)$
Word_Map.put()	$T_{12}(n)=1$	$O(1)$
Word_Map.resize()	$T_{13}(n)=n$	$O(n)$
Word_Map.putAll()	$T_{14}(n)=n$	$O(n)$
Word_Map.clear()	$T_{15}(n)=1$	$O(1)$
Word_Map.keySet()	$T_{16}(n)=1$	$O(1)$
Word_Map.values()	$T_{17}(n)=n$	$O(n)$
File_Map.size()	$T_{18}(n)=1$	$O(1)$
File_Map.isEmpty()	$T_{19}(n)=1$	$O(1)$
File_Map.containsKey()	$T_{20}(n)=n$	$O(n)$
File_Map.containsValue()	$T_{21}(n)=n$	$O(n)$
File_Map.get()	$T_{22}(n)=n$	$O(n)$
File_Map.put()	$T_{23}(n)=1$	$O(1)$
File_Map.remove()	$T_{24}(n)=1+1$	$O(1)$
File_Map.putAll()	$T_{25}(n)=n+n$	$O(n)$
File_Map.clear()	$T_{26}(n)=1$	$O(1)$
File_Map.keySet()	$T_{27}(n)=n$	$O(n)$
File_Map.values()	$T_{28}(n)=1$	$O(1)$

3 RESULT

3.1 Test Cases

I tested the program with input of various bigram words and tfidf words.

3.2 Running Results

Input: Homework PDF’s example

bigram very

tfidf coffee 0001978

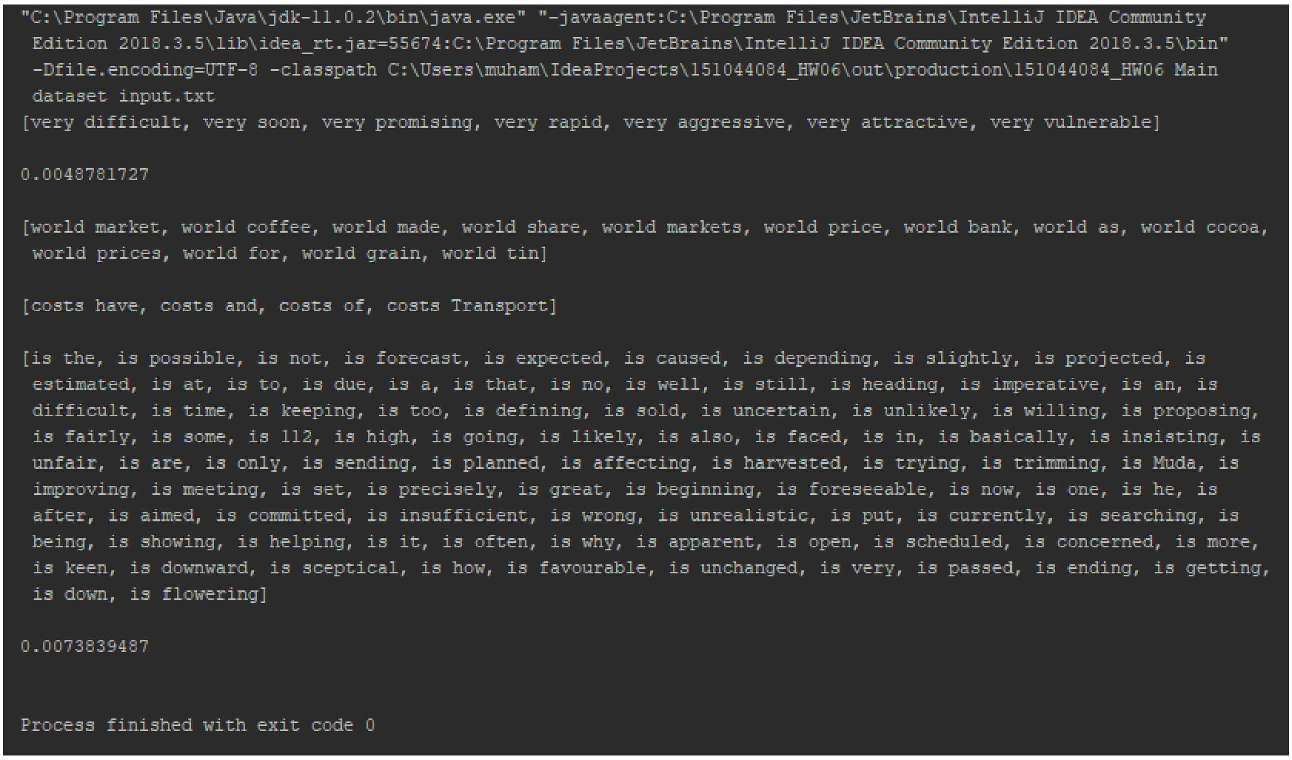
bigram world

bigram costs

bigram is

tfidf Brazil 0000178

Output:



```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
 Edition 2018.3.5\lib\idea_rt.jar=55674:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2018.3.5\bin"
 -Dfile.encoding=UTF-8 -classpath C:\Users\muham\IdeaProjects\151044084_HW06\out\production\151044084_HW06 Main
 dataset input.txt
[very difficult, very soon, very promising, very rapid, very aggressive, very attractive, very vulnerable]

0.0048781727

[world market, world coffee, world made, world share, world markets, world price, world bank, world as, world cocoa,
 world prices, world for, world grain, world tin]

[costs have, costs and, costs of, costs Transport]

[is the, is possible, is not, is forecast, is expected, is caused, is depending, is slightly, is projected, is
 estimated, is at, is to, is due, is a, is that, is no, is well, is still, is heading, is imperative, is an, is
 difficult, is time, is keeping, is too, is defining, is sold, is uncertain, is unlikely, is willing, is proposing,
 is fairly, is some, is 112, is high, is going, is likely, is also, is faced, is in, is basically, is insisting, is
 unfair, is are, is only, is sending, is planned, is affecting, is harvested, is trying, is trimming, is Muda, is
 improving, is meeting, is set, is precisely, is great, is beginning, is foreseeable, is now, is one, is he, is
 after, is aimed, is committed, is insufficient, is wrong, is unrealistic, is put, is currently, is searching, is
 being, is showing, is helping, is it, is often, is why, is apparent, is open, is scheduled, is concerned, is more,
 is keen, is downward, is sceptical, is how, is favourable, is unchanged, is very, is passed, is ending, is getting,
 is down, is flowering]

0.0073839487


Process finished with exit code 0
```

Input:

bigram isnt

tfidf world 0001978

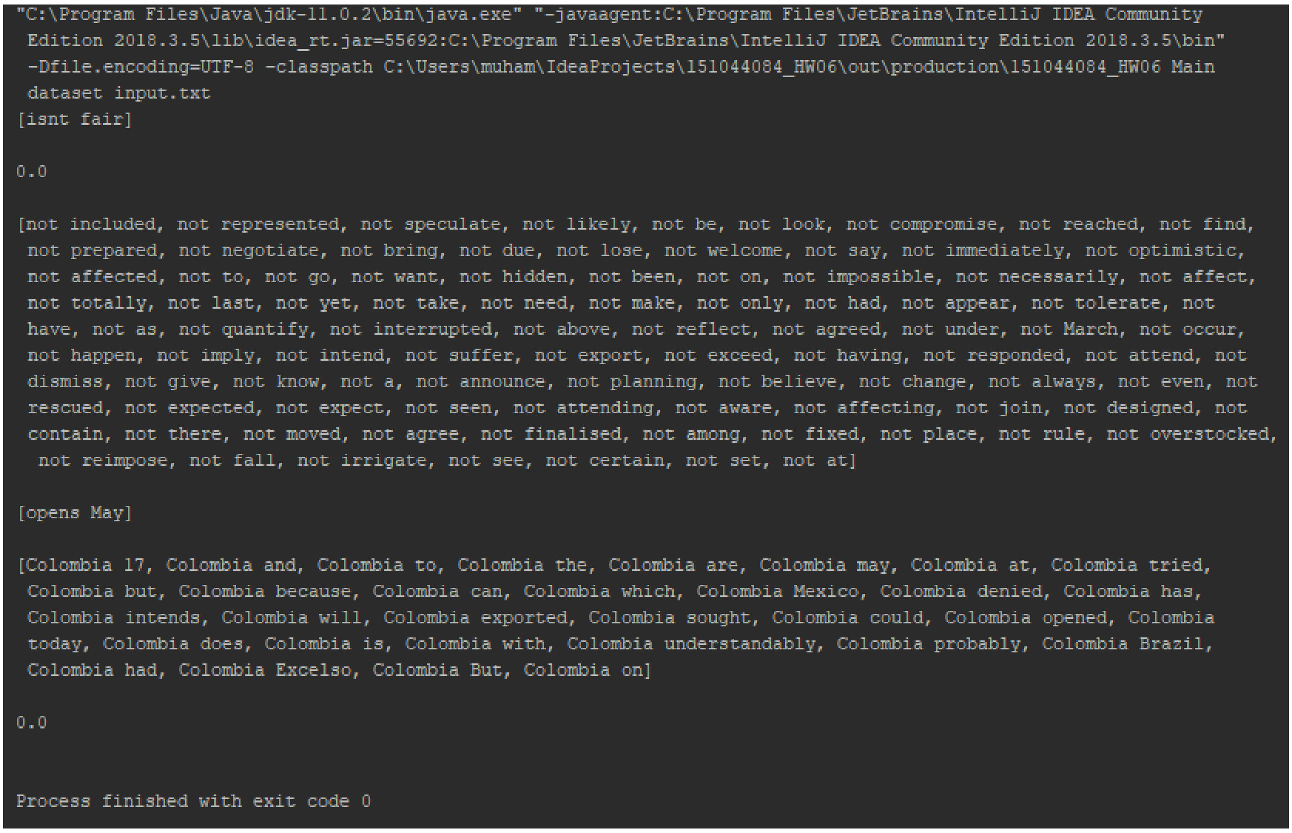
bigram not

bigram opens

bigram Colombia

tfidf costs 0000178

Output:



```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
 Edition 2018.3.5\lib\idea_rt.jar=55692:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2018.3.5\bin"
 -Dfile.encoding=UTF-8 -classpath C:\Users\muham\IdeaProjects\151044084_HW06\out\production\151044084_HW06 Main
 dataset input.txt
[isnt fair]

0.0

[not included, not represented, not speculate, not likely, not be, not look, not compromise, not reached, not find,
 not prepared, not negotiate, not bring, not due, not lose, not welcome, not say, not immediately, not optimistic,
 not affected, not to, not go, not want, not hidden, not been, not on, not impossible, not necessarily, not affect,
 not totally, not last, not yet, not take, not need, not make, not only, not had, not appear, not tolerate, not
 have, not as, not quantify, not interrupted, not above, not reflect, not agreed, not under, not March, not occur,
 not happen, not imply, not intend, not suffer, not export, not exceed, not having, not responded, not attend, not
 dismiss, not give, not know, not a, not announce, not planning, not believe, not change, not always, not even, not
 rescued, not expected, not expect, not seen, not attending, not aware, not affecting, not join, not designed, not
 contain, not there, not moved, not agree, not finalised, not among, not fixed, not place, not rule, not overstocked,
  not reimpose, not fall, not irrigate, not see, not certain, not set, not at]

[opens May]

[Colombia 17, Colombia and, Colombia to, Colombia the, Colombia are, Colombia may, Colombia at, Colombia tried,
 Colombia but, Colombia because, Colombia can, Colombia which, Colombia Mexico, Colombia denied, Colombia has,
 Colombia intends, Colombia will, Colombia exported, Colombia sought, Colombia could, Colombia opened, Colombia
 today, Colombia does, Colombia is, Colombia with, Colombia understandably, Colombia probably, Colombia Brazil,
 Colombia had, Colombia Excelso, Colombia But, Colombia on]

0.0


Process finished with exit code 0
```