

**Gebze Technical University  
Computer Engineering**

**CSE 222 - 2019 Spring**

**HOMEWORK 6 REPORT**

**AHMET MELIH YANALAK  
151044044**

Course Assistant:AYSE TURAN

# 1 INTRODUCTION

## 1.1 Problem Definition

Data is the fundamental of computers. Without data, there would be nothing much that computers could do. Nowadays, due to evolution of computers, datum that is kept by hardwares are extremely larger than before and so that keeping the datum has become demanding task. In time many data structures have been created to be used in different ways. In this project, we are asked to implement the proper data structures to keep every single word and their position in files that is inside a directory that is given. There will be 2 type of request;

1) To find all bi-grams that contains given word

2) To calculate the TFIDF value which is stand for the importance of a word in a file

Terms are explained below with details.

## 1.2 System Requirements

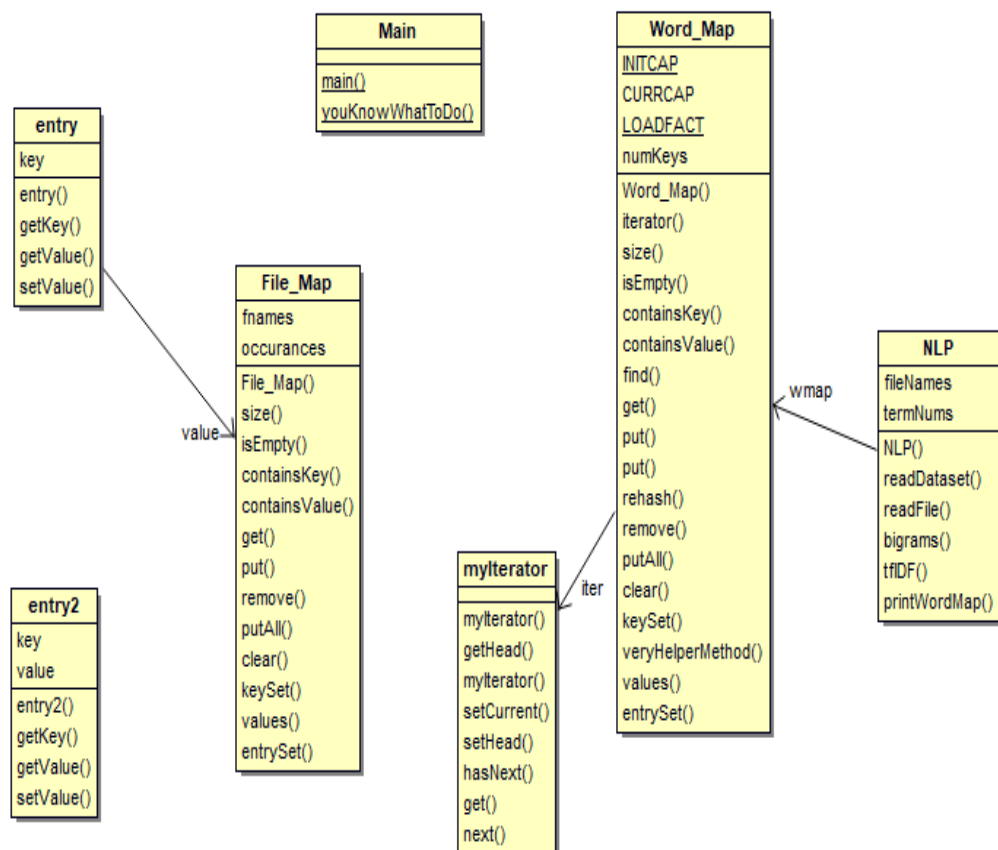
This program requires a hardware that can run an operating system which supports IntelliJ Idea Community Edition 2018 3.5 , JVM and libraries that are below:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Iterator;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
```

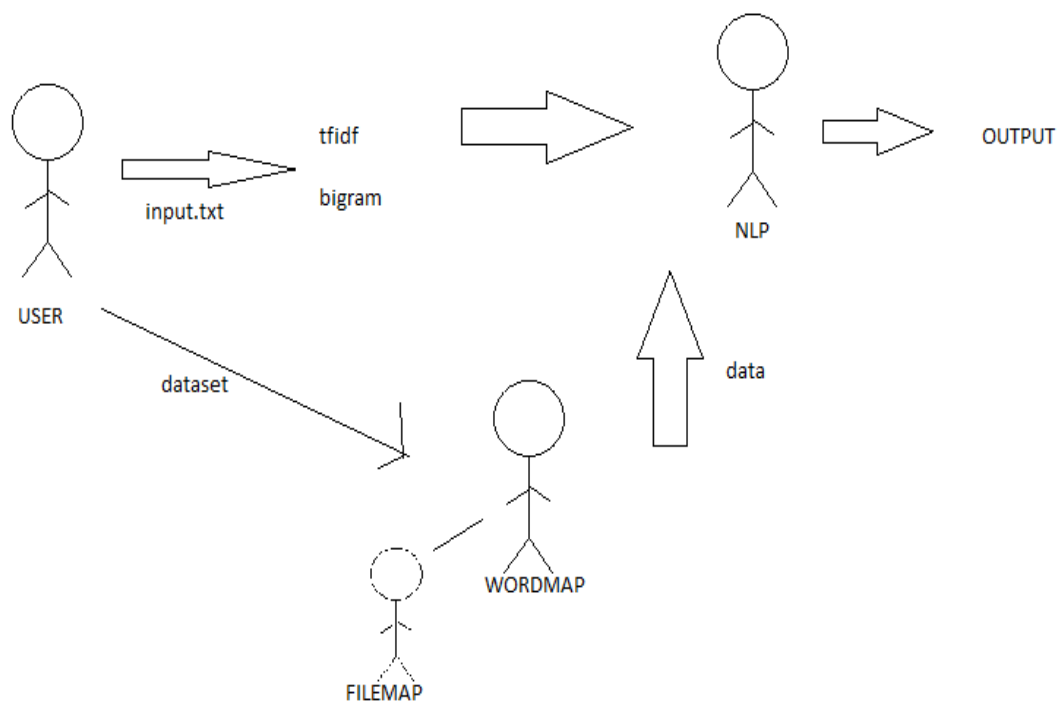
If the number of total words is considered  $N$ . Then the program uses  $4N$  amount of memory. Because, for each word, indexes should be kept and also each node should have a next Node.

## 2 METHOD

### 2.1 Class Diagrams



### 2.2 Use Case Diagrams



User creates an input file to show what is requested, and also show the path of the directory which contains dataset. While nlp doing tasks, it uses dataset that is kept in wordmap.

```
Map<Integer,String> map = Map.of(1, "A", 2, "B", 3, "C");
```



key	value
1	A
2	B
3	C

## 2.3 Problem Solution Approach

In order to achieve this task, I implemented HashMap whose key type is word(String) and value type is FileMap which keeps the occurrences of the word in different files. First of all, the files that is given by user must be processed to keep dataset in the HashMap. While processing files, each word should have been added separately. In order to achieve this task, Put method of HashMap is implemented specifically. The number of the total words and file numbers also counted to calculate tfidf value which is easier part. To achieve finding all bigrams, I implemented helper method which takes file name and index parameter and finds the word in that position. After taking word parameter in bigram method,, helper method finds all the words that is adjacent to parameter word and puts them to a list in a loop.

Here is the complexities of Methods:

WordMap:

Iterator ->  $O(1)$

Size ->  $O(1)$

IsEmpty ->  $O(1)$

containsKey ->  $O(n)$  (in case of collision)  $O(1)$  best case

containsValue ->  $O(n)$  at the worst case but since the iterator used, it will be more efficient

get ->  $O(n)$  (in case of collision)  $O(1)$  best case

put ->  $O(n)$  (in case of collision and rehashing)  $O(1)$  best case

rehash ->  $O(n)$  worst and best

putAll ->  $O(n)$

keySet ->  $O(n)$

values ->  $O(n)$  Since they copy the data its linear complexity

entrySet ->  $O(n)$

veryHelperMethod ->  $O(n^2)$  (checks if it contains a key in a while loop)

FileMap:

Size ->  $O(1)$

isEmpty ->  $O(1)$

containsKey ->  $O(n)$  (searching in underlying array of arraylist)

containsValue ->  $O(n)$  (searching in underlying array of arraylist)

get ->  $O(n)$  (finding index)

put ->  $O(n)$  (controls if it contains the key)

putAll ->  $O(n)$

keySet ->  $O(n)$

values ->  $O(n)$  Since they copy the data its linear complexity

entrySet ->  $O(n)$

NLP:

Bigrams ->  $O(n^4)$  -> (In 2 nested loop helper method is called)

TFIDF ->  $O(n)$  -> get method is called for wordmap

PrintWordMap ->  $O(n^2)$  -> 2 nested loops for printing

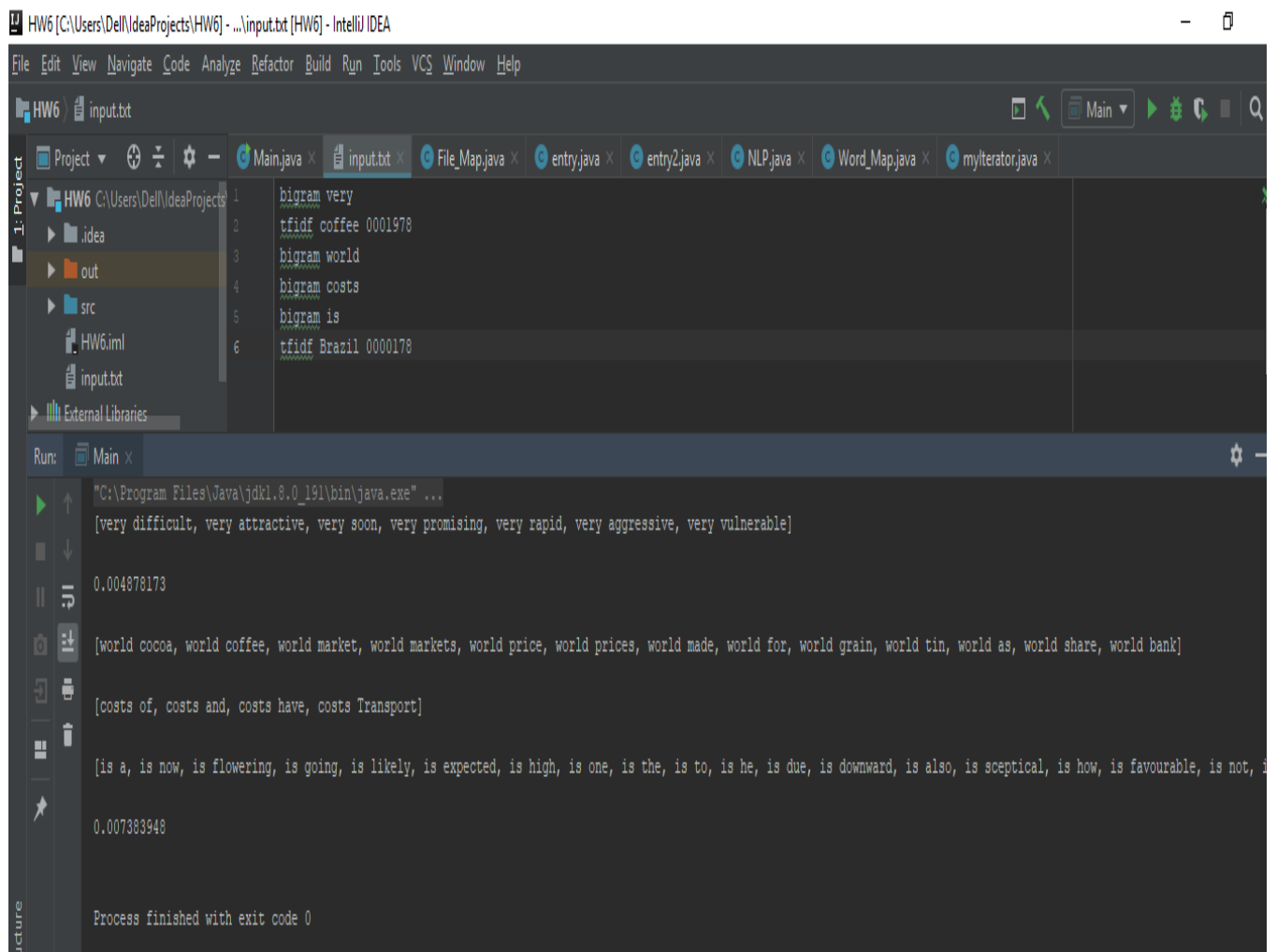
ReadDataSet ->  $O(n^2)$  -> calls contains method inside 2 nested loops

## 3 RESULT

### 3.1 Test Cases

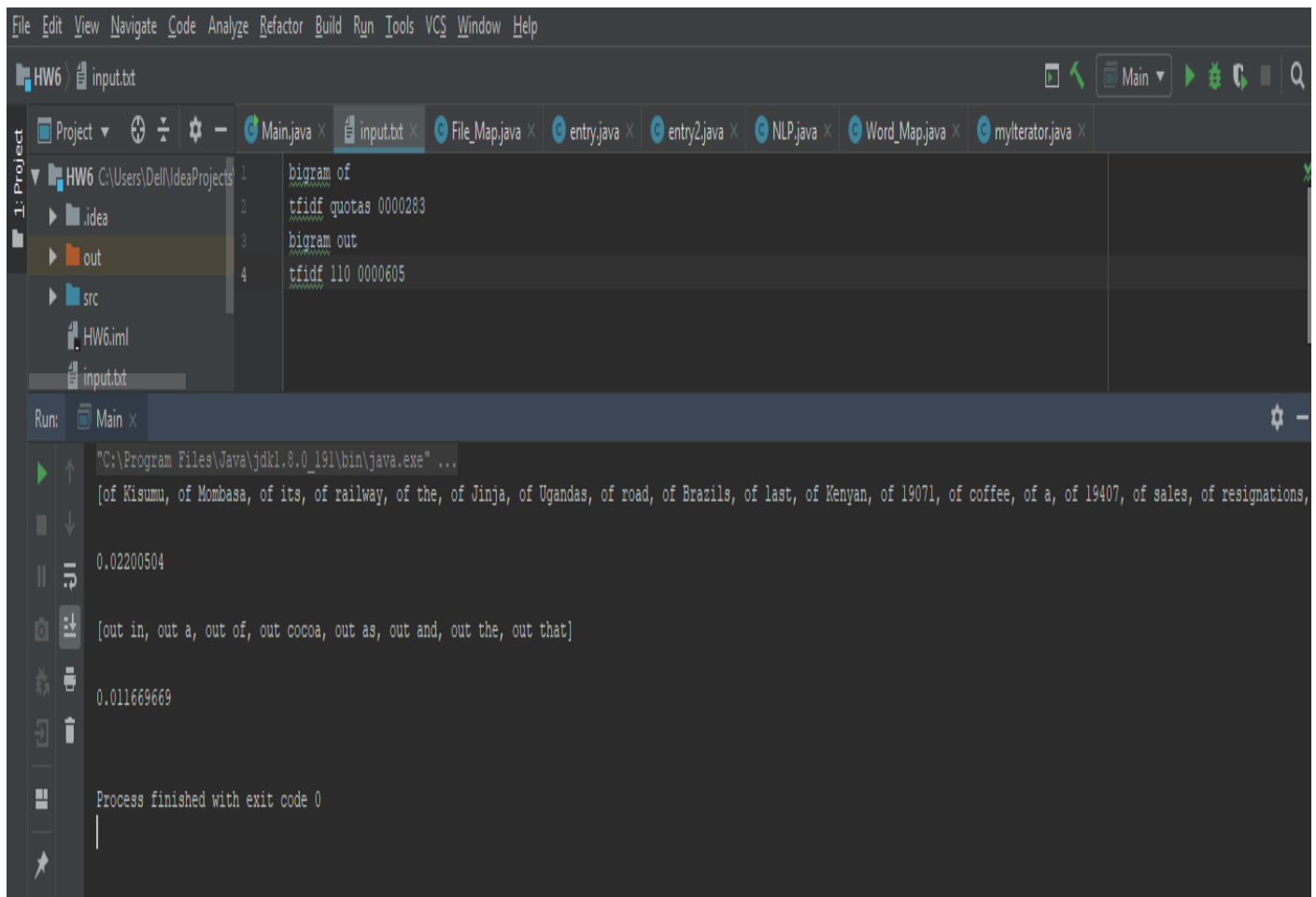
The program is checked for several inputs, which contains different input files, different directories and also different datasets.

### 3.2 Running Results



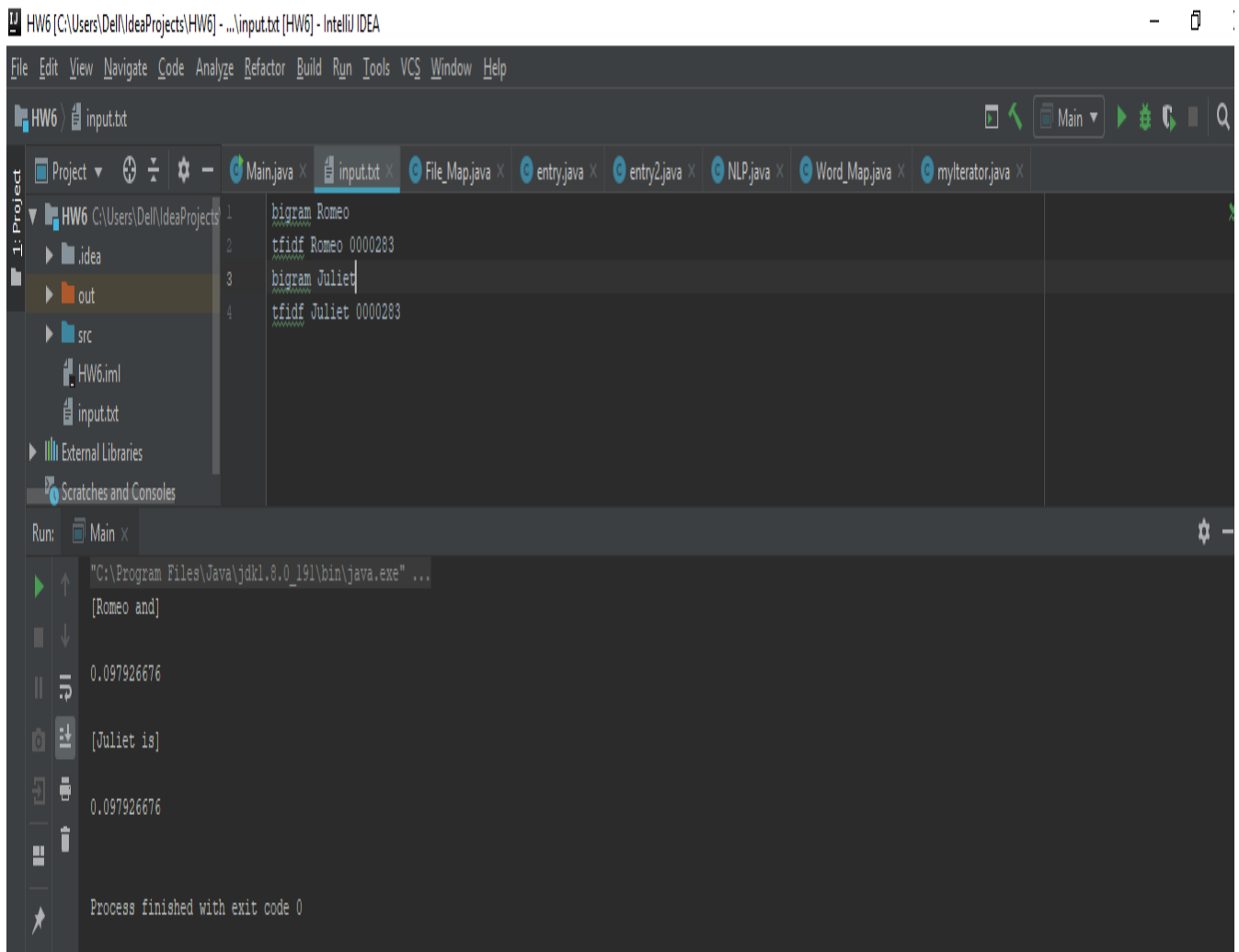
```
HW6 [C:\Users\De\IdeaProjects\HW6] - ..\input.txt [HW6] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HW6 input.txt
Project HW6 C:\Users\De\IdeaProjects
  .idea
  out
  src
  HW6.iml
  input.txt
  External Libraries
1 bigram very
2 tfidf coffee 0001978
3 bigram world
4 bigram costs
5 bigram is
6 tfidf Brazil 0000178
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
[very difficult, very attractive, very soon, very promising, very rapid, very aggressive, very vulnerable]
0.004878173
[world cocoa, world coffee, world market, world markets, world price, world prices, world made, world for, world grain, world tin, world as, world share, world bank]
[costs of, costs and, costs have, costs Transport]
[is a, is now, is flowering, is going, is likely, is expected, is high, is one, is the, is to, is he, is due, is downward, is also, is sceptical, is how, is favourable, is not, is]
0.007383948
Process finished with exit code 0
```

It is the output with standart input file that is given by example.



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HW6 input.txt
Project HW6 C:\Users\Del\IdeaProjects
  .idea
  out
  src
  HW6.iml
  input.txt
Main.java x input.txt x File_Map.java x entry.java x entry2.java x NLP.java x Word_Map.java x myIterator.java x
1 bigram of
2 tfidf quotas 0000283
3 bigram out
4 tfidf 110 0000605
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
[of Kisumu, of Mombasa, of its, of railway, of the, of Jinja, of Ugandas, of road, of Brazils, of last, of Kenyan, of 19071, of coffee, of a, of 19407, of sales, of resignations,
0.02200504
[out in, out a, out of, out cocoa, out as, out and, out the, out that]
0.011669669
Process finished with exit code 0
```

With Different input File



With different dataset