Gebze Technical University
Computer Engineering

CSE 222 – 2019 Spring

HOMEWORK 6 REPORT

CANER KARAKAŞ
131044061

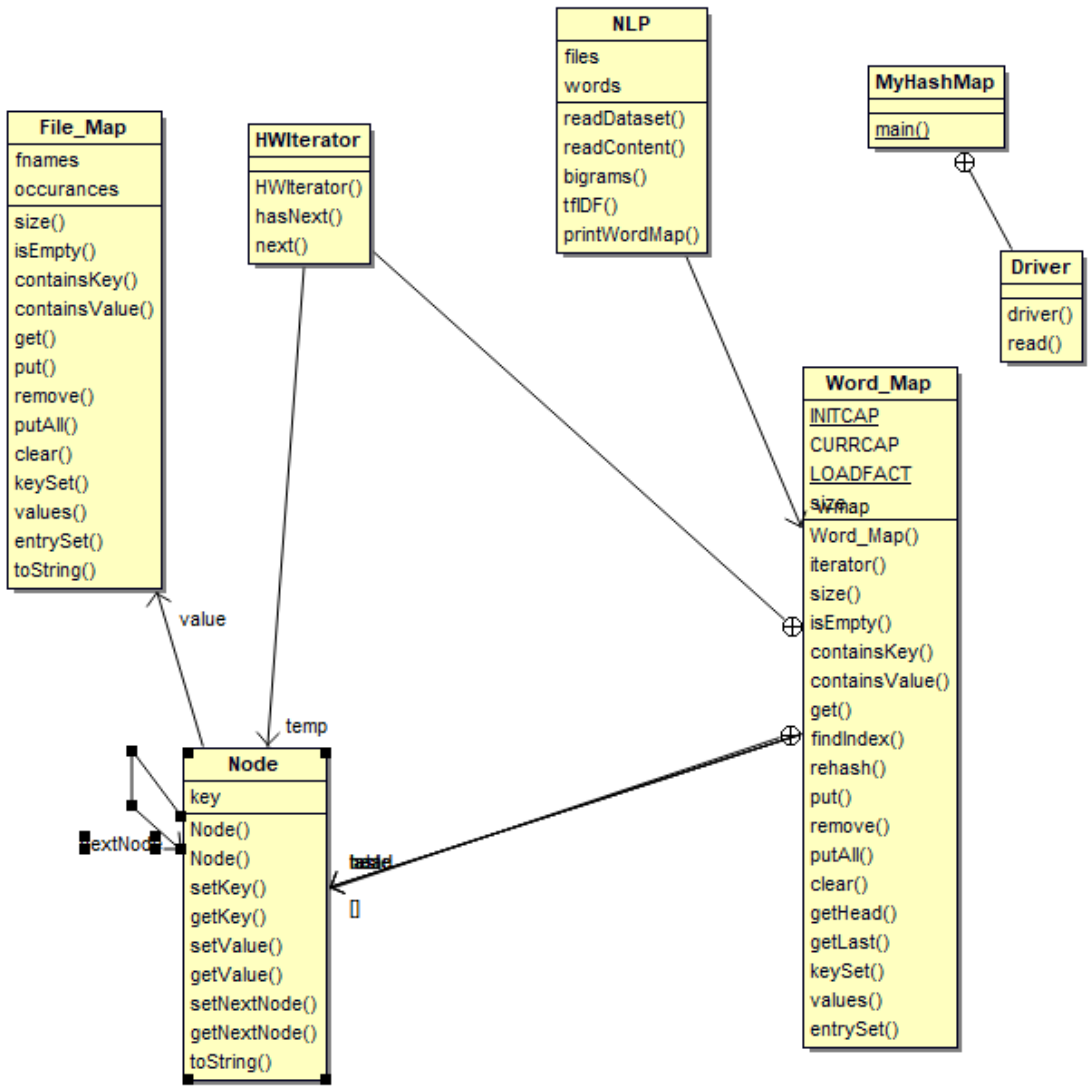
INTRODUCTION

1.1 Problem Definition

In this homework, we are asked to read the different files. We read them, store words. Then, put in wordmap class with the filemap class to help this class, we store the words based on their location in the files. Our goal is to perform bigram and TFIDF operations according to these words, perform this with the map interface.

2 METHOD

2.1 Class Diagrams



2.2 Problem Solution Approach

-File_Map Class

This class retrives fnames and occurances arraylists as class members. At the same time, it takes implements from the Map interface. Therefore, it override some methods. These methods are as follows size, isEmpty, containsKey, containsValue, get, put, remove, putAll, clear, keySet, values, entrySet. File_Map Class exists to keep the words in the other class to be kept which file and in which position. There are fnames for file names, there are also occurances for locations.

The containsKey method checks whether the given key is in the fnames. Arraylist's containsKey method is called. The containsValue method checks whether the given value is in the occurance. It was made in the same way as the containsKey method.

The put method is used to add the given key and the value to the data. first checks whether the given key is in the data. If not, it inserts into the data. If so, adds the new coordinate and continues.

-Word_Map Class

This class implements from the Map interface and Iterable interface. It holds INITCAP, CURRCAP, LOADFACT, table, head, last, size class members. It also has Node class and HWIterator class. This class also override the same functions as File_Map class.

It connects all data with nodes. It also holds the table array. It also uses the hash mat when holding the array. Rehash and findIndex methods have been written to create Hash logic. The Rehash method is necessary to create new space where capacity is not sufficient. new openings were made by 2 * n + 1 process. The FindIndex method is required to find locations where the hash codes overlap or where they do not overlap. If the index is negative, it is blocked. If the space produced is full, one side is rotated.

The Node class holds the key and the File_Map value.

-NLP Class

In this class, the information and the words of all the files to be read are kept. All the words read are stored to the wmap, which is a Word_Map object.

In each file of the given word, Bigram keeps and returns a list with its neighbor. first, it checks the given word whether the data is in it. After that, Bigrams were found using treeset and arraylist structures.

3 RESULT

Tests

```
bigram very
tfidf coffee 0001978
bigram world
bigram costs
bigram is
tfidf Brazil 0000178
```

Running Results

```
----------------------
[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 coffee, world markets, world market, world price, world mar

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

[is the, is possible, is not, is forecast, is forecast, is expected, is caused, is depending, is expected, is expected,

0.0073839487
```

```
proposed
[0000048, 0000165, 0000527, 0000677, 0001004, 0001155, 0001327, 0002827, 0007882, 0008477]
[[49, 61], [210], [318], [34], [135, 162], [31], [55], [109], [656], [297]]
one
[0000048, 0000165, 0000283, 0000605, 0000677, 0000764, 0000783, 0000828, 0001004, 0001085, 0001184, 0001234, 000132
[[57], [382], [94], [66], [89], [90], [22], [67, 325, 573, 838], [306], [35], [389, 422], [63], [232], [298, 934],
shareout
[0000048]
[[62]]
still
[0000048, 0000283, 0000402, 0000605, 0005012, 0006272, 0007218, 0007320, 0007660, 0007709, 0007882, 0008364, 000842
[[64], [18], [24], [202], [131], [22, 135], [139, 269], [139], [72], [173, 184], [1170], [129], [129], [191, 320, 4
include
[0000048, 0000167, 0004598, 0007218, 0007985]
[[65], [181], [79], [313, 392, 408], [96]]
declarations
[0000048]
[[67]]
Talks
[0000165, 0000202, 0000402, 0006272, 0008350, 0008606, 0008656]
[[0], [0], [0], [91], [70], [24], [64]]
possibility
[0000165, 0008477, 0008911]
[[3], [144], [17]]
global
[0000165, 0000527, 0001155, 0007218, 0008477]
[[6], [288], [98], [405], [77]]
coffee
[0000165, 0000167, 0000178, 0000194, 0000202, 0000458, 0000503, 0000527, 0000605, 0000639, 0000641, 0000677, 000075
[[7, 55, 75, 84, 117, 394, 412, 424], [283], [47, 133], [14, 43, 70, 105], [2], [6, 15], [17, 66, 88], [12, 22, 32,
have
[0000165, 0000458, 0000527, 0000605, 0000639, 0000677, 0000783, 0000828, 0001004, 0001085, 0001155, 0001180, 000118
[[10, 185, 289, 418, 427], [25], [406], [229, 246], [290], [109], [3], [3, 222, 230, 276, 750], [175, 493], [61, 12
extended
[0000165, 0000178, 0000402, 0001085]
[[12], [14], [3], [81]]
into
[0000165, 0000178, 0000639, 0000764, 0000828, 0001603, 0003805, 0005750, 0007218, 0007882, 0008304, 0008656]
[[13, 331], [18], [29], [129], [700], [367, 700], [494], [336], [443], [281], [19], [279, 544]]
today
[0000165, 0000202, 0000402, 0000527, 0000639, 0000764, 0000828, 0001184, 0001603, 0001978, 0003195, 0003558, 000552
[[14], [19], [137], [31], [133], [156], [55], [2], [432], [234], [318], [36, 86], [12, 58], [31], [35], [39], [462
sparks
```

TIME COMPLEXITY

Class	Method	Complexty
File_Map	size()	O(1)
File_Map	get()	O(n)
File_Map	put()	O(n)
File_Map	containsKey()	O(n)
File_Map	putAll()	O(n)
File_Map	clear()	O(n)
File_Map	keySet()	O(n)
File_Map	values()	O(n)
File_Map	containValues()	O(n)
File_Map	entrySet()	O(n)
File_Map	values()	O(n)
Word_Map	get()	O(1)
Word_Map	put()	O(1)
Word_Map	containKey()	O(1)
Word_Map	putAll()	O(n)
Word_Map	keySet()	O(n)

Word_Map	values()	O(n)
Word_Map	containValues ()	O(n)
Word_Map	entrySet ()	O(n)
Word_Map	rehash()	O(n)
Word_Map	findIndex()	O(1)
NLP	readDataSet()	O(n)
NLP	readContent()	O(n)
NLP	tfIDF()	O(n)
NLP	printWordMap()	O(n)