Advanced Data Structure Project 1

Mehmet MUM 150114051

1-) Structure

I have chosen compressed trie structure for this project. In the queries we do not search for substring, we are searching for words. So, we do not need suffix links and output links. For space efficiency, I did not choose trie. We can easily convert a trie to compressed trie by combining internal nodes which has an only one edge with it's child. In addition, I did not choose suffix tree because as I said before we are looking for words, there will be waste of space in suffix tree.

- **Inside Nodes**

  In order to make queries faster, I have stored several data inside nodes:

  - **private boolean accept:** this variable stores the node is accepting the word or not.
  - **private String str:** this variable stores node's chars.
  - **private String[][] accepted_files:** this variable stores which files has the word and store the word's index in that file.
  - **private Node children[]:** this variable stores children of the node.

2-) Preprocessing

In the preprocessing phase, program takes a directory from user. Creates a global trie. For each txt file in the directory:

- Program takes the content of txt file as a string.
- First removes \t \n \r \b \f \' \" characters.
- Then, replaces punctuation with blank character. (except " and ')
- In order to get words, split the string with blank char.
- Inserts each word to the trie.
- After finish inserting words to trie, program compress the trie to create compressed trie.

The first character's index of a text file begins with 1. The indexes of words in the file and in the nodes may not be same. Because, program is removing some special characters (\t \n \r \b \f \' \"). So, some of the indexes are changing. Program prints a word's index in the string which some characters may be removed.

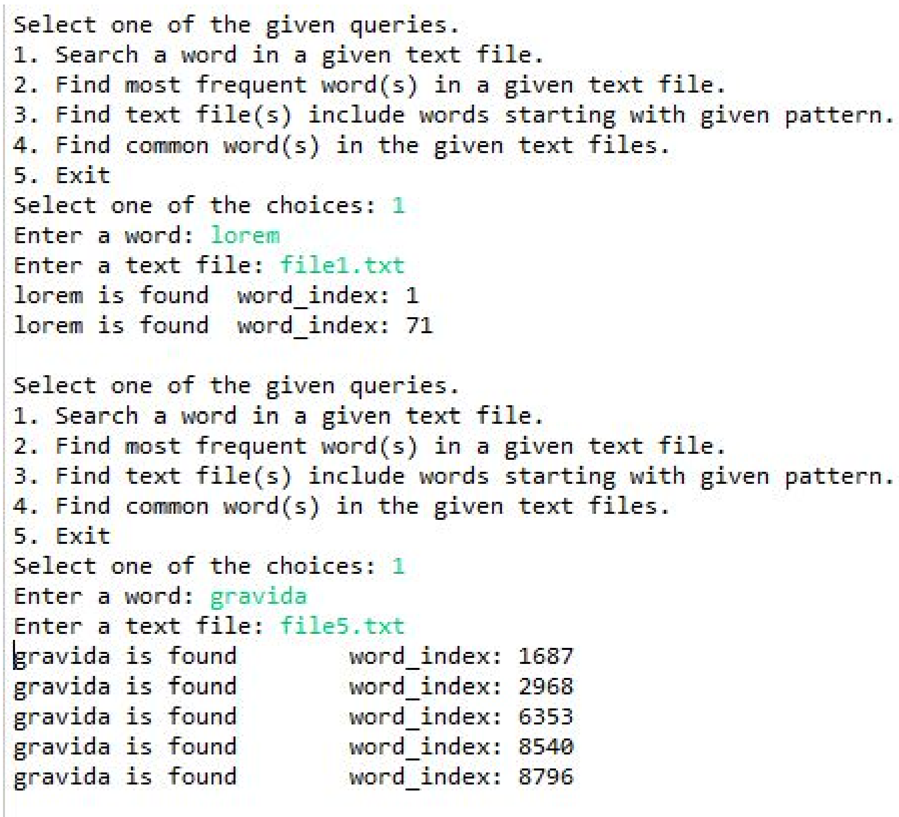
3-) Query

There are four possible queries that user can perform.

1) Search a word in a given text file.
2) Find most frequent word(s) in a given text file.
3) Find text file(s) include words starting with given pattern.
4) Find common word(s) in the given text files.

3.1) Search a word in a given text file

In this query, program searches given word in the tire. In order to search in the compressed trie, program walk down by comparing characters of nodes with characters of word (from beginning of the word). If they are equal then program get rid of matching part from word and continues to search until stuck or word becomes empty word. If it stucks in the compressed trie, that means the word is not found. If word becomes empty word, that means the word has been found. Then check for the node is accepted or not. If it is accepted, that means the word is in the compressed trie. If the word is in the compressed trie, program checks the given file has the word or not by checking *accepted_files* variable. If the word exists in the file then program writes all positions with the word as output.



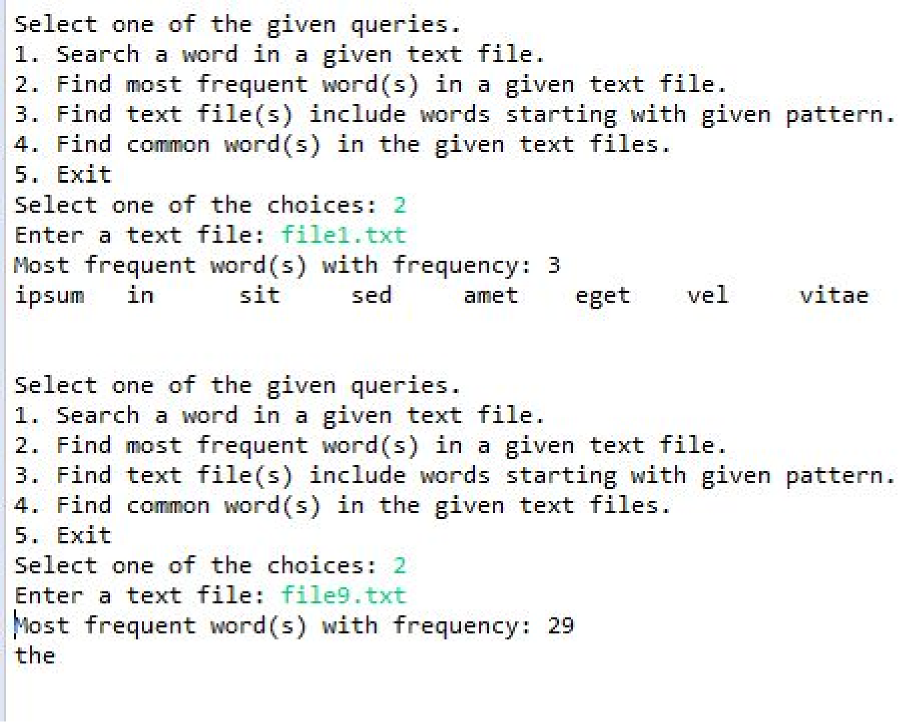
```
Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 1
Enter a word: lorem
Enter a text file: file1.txt
lorem is found  word_index: 1
lorem is found  word_index: 71

Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 1
Enter a word: gravida
Enter a text file: file5.txt
gravida is found        word_index: 1687
gravida is found        word_index: 2968
gravida is found        word_index: 6353
gravida is found        word_index: 8540
gravida is found        word_index: 8796
```

In the first part, program searches for word “lorem” in the file “file1.txt” and finds 2 words and prints both of them with word index. In the second part, program prints all “gravida” words with their indexes in the “file5.txt”.

3.2) Find most frequent word(s) in a given text file

In this query, program finds most frequent word(s) in a given text file. In order to do that, program looks every node in the compressed trie. If a node is accepted node, then program calculates number of the found word in the given file by looking into *accepted_files* variable. Variable *frequency* stores number of repetition of most frequent word(s) and *frequent_words* variable stores most frequent word(s). If the count is bigger than the *frequency*, *frequency* becomes the count and program removes all data in the *frequent_word* then inserts the word into it. If count is equal *frequency*, program just inserts the word into *frequent_words*. If count is less than *frequency*, program does nothing. After finish tracing the tree, program prints every word in the *frequent_words*.



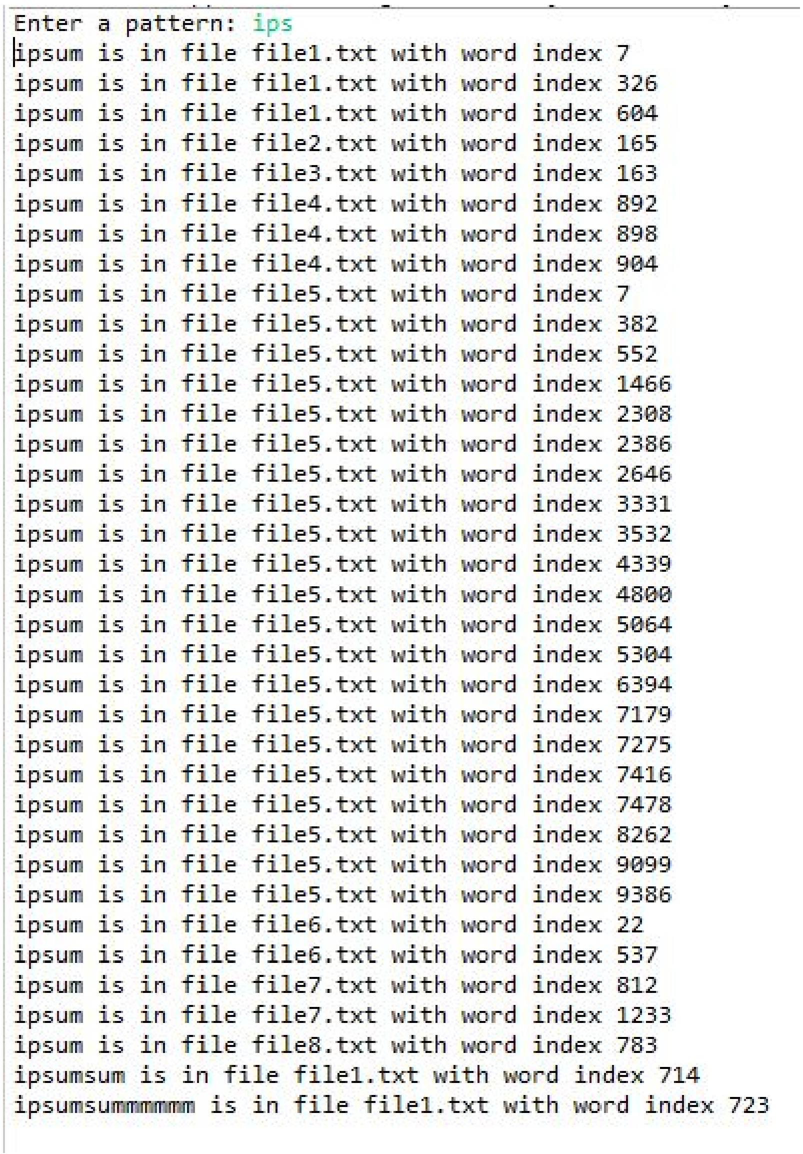
```
Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 2
Enter a text file: file1.txt
Most frequent word(s) with frequency: 3
ipsum   in      sit     sed     amet    eget    vel     vitae


Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 2
Enter a text file: file9.txt
Most frequent word(s) with frequency: 29
the
```

In the first part, program finds most frequent word(s) in the “file1.txt”. Program finds 7 words, these are “ipsum”, “in”, “sit”, “sed”, “amet”, “eget”, “vel”, “vitae” with frequency 3. In the second part, program finds “the” as most frequent word with frequency 29 in the “file9.txt”.

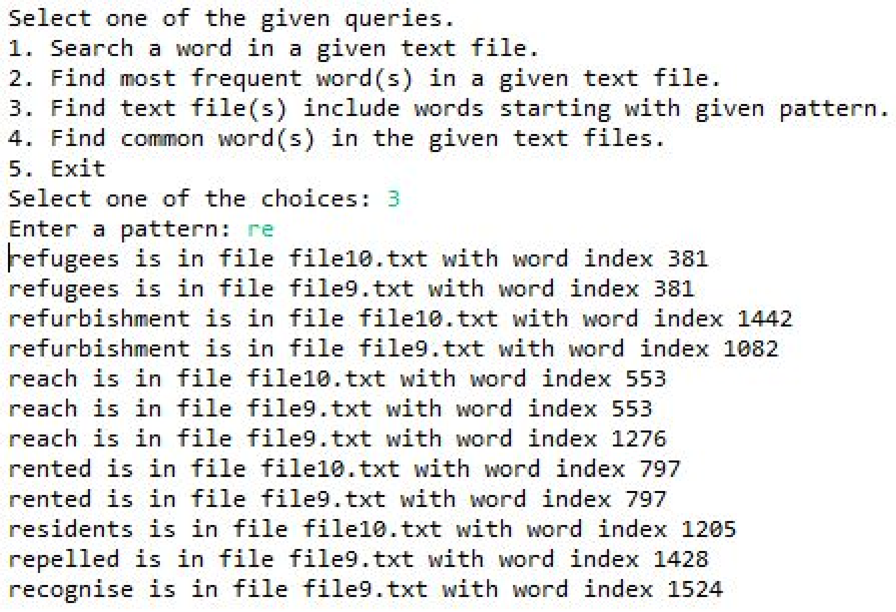
3.3) Search a word in a given text file

In this query, program finds all words starting with given pattern and prints them with their file name and their indexes. In order to do that, program search for pattern. Once the program find the given pattern, it looks the node and every words under the node.



```
Enter a pattern: ips
ipsum is in file file1.txt with word index 7
ipsum is in file file1.txt with word index 326
ipsum is in file file1.txt with word index 604
ipsum is in file file2.txt with word index 165
ipsum is in file file3.txt with word index 163
ipsum is in file file4.txt with word index 892
ipsum is in file file4.txt with word index 898
ipsum is in file file4.txt with word index 904
ipsum is in file file5.txt with word index 7
ipsum is in file file5.txt with word index 382
ipsum is in file file5.txt with word index 552
ipsum is in file file5.txt with word index 1466
ipsum is in file file5.txt with word index 2308
ipsum is in file file5.txt with word index 2386
ipsum is in file file5.txt with word index 2646
ipsum is in file file5.txt with word index 3331
ipsum is in file file5.txt with word index 3532
ipsum is in file file5.txt with word index 4339
ipsum is in file file5.txt with word index 4800
ipsum is in file file5.txt with word index 5064
ipsum is in file file5.txt with word index 5304
ipsum is in file file5.txt with word index 6394
ipsum is in file file5.txt with word index 7179
ipsum is in file file5.txt with word index 7275
ipsum is in file file5.txt with word index 7416
ipsum is in file file5.txt with word index 7478
ipsum is in file file5.txt with word index 8262
ipsum is in file file5.txt with word index 9099
ipsum is in file file5.txt with word index 9386
ipsum is in file file6.txt with word index 22
ipsum is in file file6.txt with word index 537
ipsum is in file file7.txt with word index 812
ipsum is in file file7.txt with word index 1233
ipsum is in file file8.txt with word index 783
ipsumsum is in file file1.txt with word index 714
ipsumsummmmmm is in file file1.txt with word index 723
```

In this query, the pattern is “ips” and program has found all words starting with the given pattern. These are “ipsum”, “ipsumsum”, “ipsumsummmmmmm”. There are 3 “ipsum” in the file1.txt program has printed all of them with their indexes.

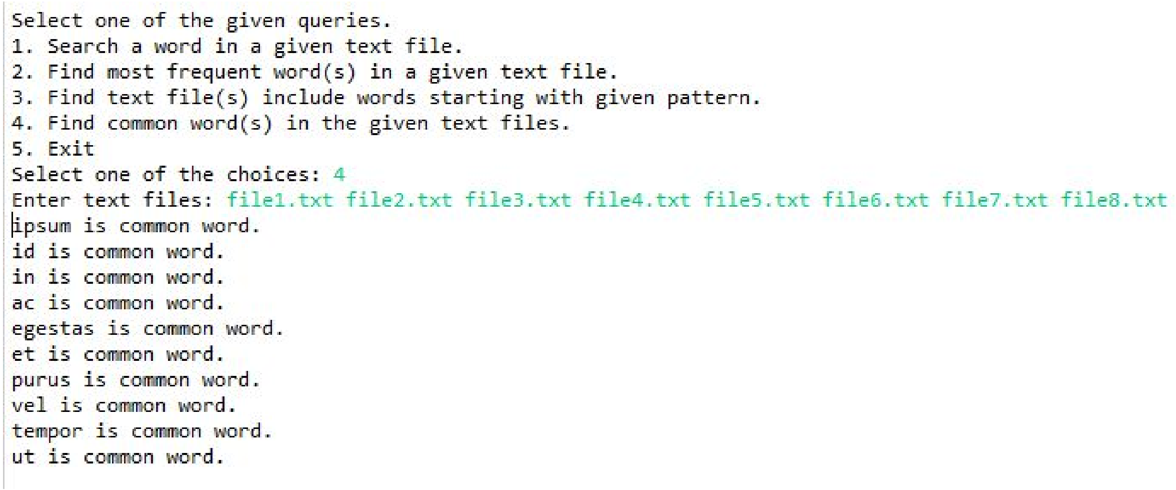


```
Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 3
Enter a pattern: re
refugees is in file file10.txt with word index 381
refugees is in file file9.txt with word index 381
refurbishment is in file file10.txt with word index 1442
refurbishment is in file file9.txt with word index 1082
reach is in file file10.txt with word index 553
reach is in file file9.txt with word index 553
reach is in file file9.txt with word index 1276
rented is in file file10.txt with word index 797
rented is in file file9.txt with word index 797
residents is in file file10.txt with word index 1205
repelled is in file file9.txt with word index 1428
recognise is in file file9.txt with word index 1524
```

In this query, the pattern is “re” and program has found all words starting with the given pattern. These are “refugees”, “refurbishment”, “reach”, “rented”, “residents”, “repelled”, “recognise”.

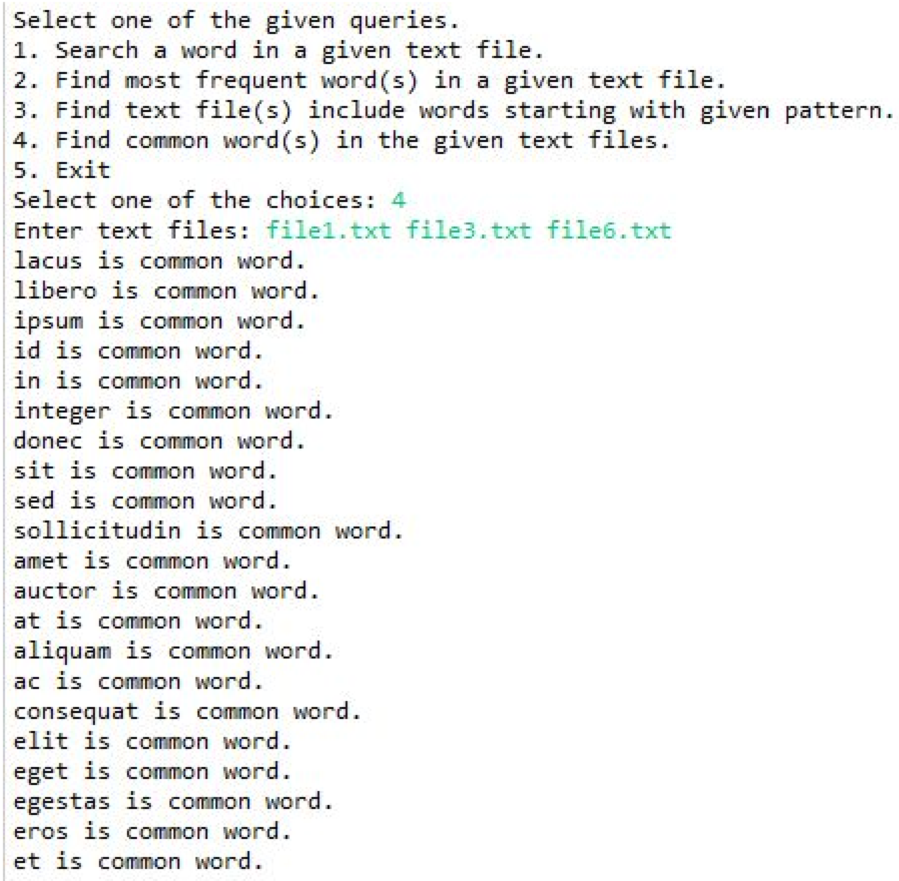
3.4) Find common word(s) in the given text files.

In this query, program finds and prints all common word(s) in given files. In order to do that, program looks every node in the compressed trie. If the node is accepted, then looks into *accepted_files* variable. If all given files exist in the variable, then the word is common word in the text files.

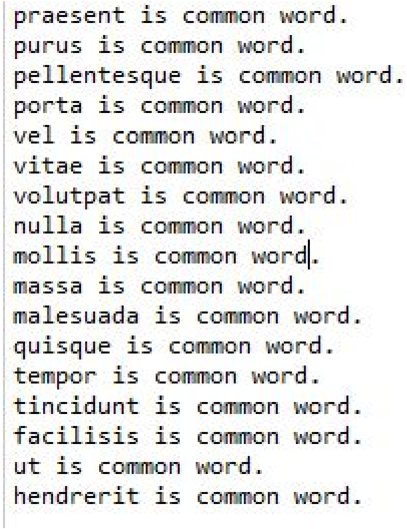


```
Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 4
Enter text files: file1.txt file2.txt file3.txt file4.txt file5.txt file6.txt file7.txt file8.txt
ipsum is common word.
id is common word.
in is common word.
ac is common word.
egestas is common word.
et is common word.
purus is common word.
vel is common word.
tempor is common word.
ut is common word.
```

There are 10 common words in the files “file1.txt”, “file2.txt”, “file3.txt”, “file4.txt”, “file5.txt”, “file6.txt”, “file7.txt”, “file8.txt”.



```
Select one of the given queries.
1. Search a word in a given text file.
2. Find most frequent word(s) in a given text file.
3. Find text file(s) include words starting with given pattern.
4. Find common word(s) in the given text files.
5. Exit
Select one of the choices: 4
Enter text files: file1.txt file3.txt file6.txt
lacus is common word.
libero is common word.
ipsum is common word.
id is common word.
in is common word.
integer is common word.
donec is common word.
sit is common word.
sed is common word.
sollicitudin is common word.
amet is common word.
auctor is common word.
at is common word.
aliquam is common word.
ac is common word.
consequat is common word.
elit is common word.
eget is common word.
egestas is common word.
eros is common word.
et is common word.
```



```
praesent is common word.
purus is common word.
pellentesque is common word.
porta is common word.
vel is common word.
vitae is common word.
volutpat is common word.
nulla is common word.
mollis is common word.
massa is common word.
malesuada is common word.
quisque is common word.
tempor is common word.
tincidunt is common word.
facilisis is common word.
ut is common word.
hendrerit is common word.
```

These are the common words in the files “file1.txt”, “file3.txt”, “file6.txt”.