



CSE 4094 Special Topics in Computer Engineering

Advanced Data Structures

Project 2 - Mini Search Engine

Report

Group Members:

Ömer Faruk Çakı – 150117821

Buğra Akdeniz – 150116072

Havva Karaçam – 150315029

Course Instructor:

Fatma Corut Ergin

Introduction

In this project we made a mini search engine using C++ programming language. We implemented that using Generalized Suffix Tree data structure.

We choosed generalized suffix tree this seem as the best one for us considering we are gonna put many words on the tree. We added tags to leaf nodes(end of word nodes), by doing that we can know in which files that words belongs to. Also information related the word's position on that file is included.

Structures

Here we have a struct named `locationInfo`. This is the information tag available on each leaf/end nodes. As we mentioned above, it contains the name of the file which the word is belongs to alongside with corresponding line number and position on that file.

```
struct locationInfo {  
    std::string file;  
    int lineNumber;  
    int index;  
};
```

Here we have a `node` structure. This is the most important component of this project. Also, we decided to keep every character as a single node and do not combine silly nodes, because we find that easier to implement. That is why we used `char` to keep a single character. The Boolean `isLeaf` represents whether it is a leaf node or not. `locations` variable on the other hand, which is vector of `locationInfo` we discussed above, holds information about the words and only available in leaf nodes. Lastly, `childs` array holds the pointer addresses of child

nodes (if any), so we can iterate through child nodes. 26 for English letters, 10 for digits and 1 for leaf node pointer, remaining are extra and not used. Using an array here give us an ability to $O(1)$ access while inserting nodes.

```
struct node {  
    char c; // character, '\0' for null terminator  
    bool isLeaf;  
    std::vector<locationInfo *> locations;  
    node *childs[40];  
};
```

Functions

Here we will discuss considerable functions of the program. We may not mention some of them here in case of simplicity of this report since they are helper functions and responsible with simple task which is not directly related with this project and most already commented on source codes.

While building the Suffix Tree, all special characters other than alphabet characters and numbers considered as space (skipped). For example 'co-founder' handled as two different word like 'co' and 'host'.

main()

Starting point of the program, reads comand line parameters. The first and only needed parameter is the path of the folder which contains text files to be read. It reads all the files in provided directory and construct a single suffix tree with all the words on them. And we also have a promth to ask users which option they want to execute. We have defined the root node inside main.

insert()

This function is used to insert a word into the tree. It traverses the all the characters and insert them into tree if not available, iterates over the next one. After each word, we insert a leaf node with related informations we discussed above.

search()

This function is used to search a prefixes. It searches the tree for the given query and if it is found, prints informations of results such as in which file that is contained and at which location etc using Depth First Search (DFS).

printCommons()

This function takes a list of file names, and it finds all common words inside these files. In order to find common words we followed these steps:

1. Traverse to all leaf nodes using DFS
2. Check tags to find out whether the word is available in all the files which are provided into this function by comparing two vectors.
3. If the current word is included in all the files in the list, that means it's a common word between files which we selected.

Compile Instructions

Source code can be compiled with the command below, c++17 is required since the program has some c++17 features.

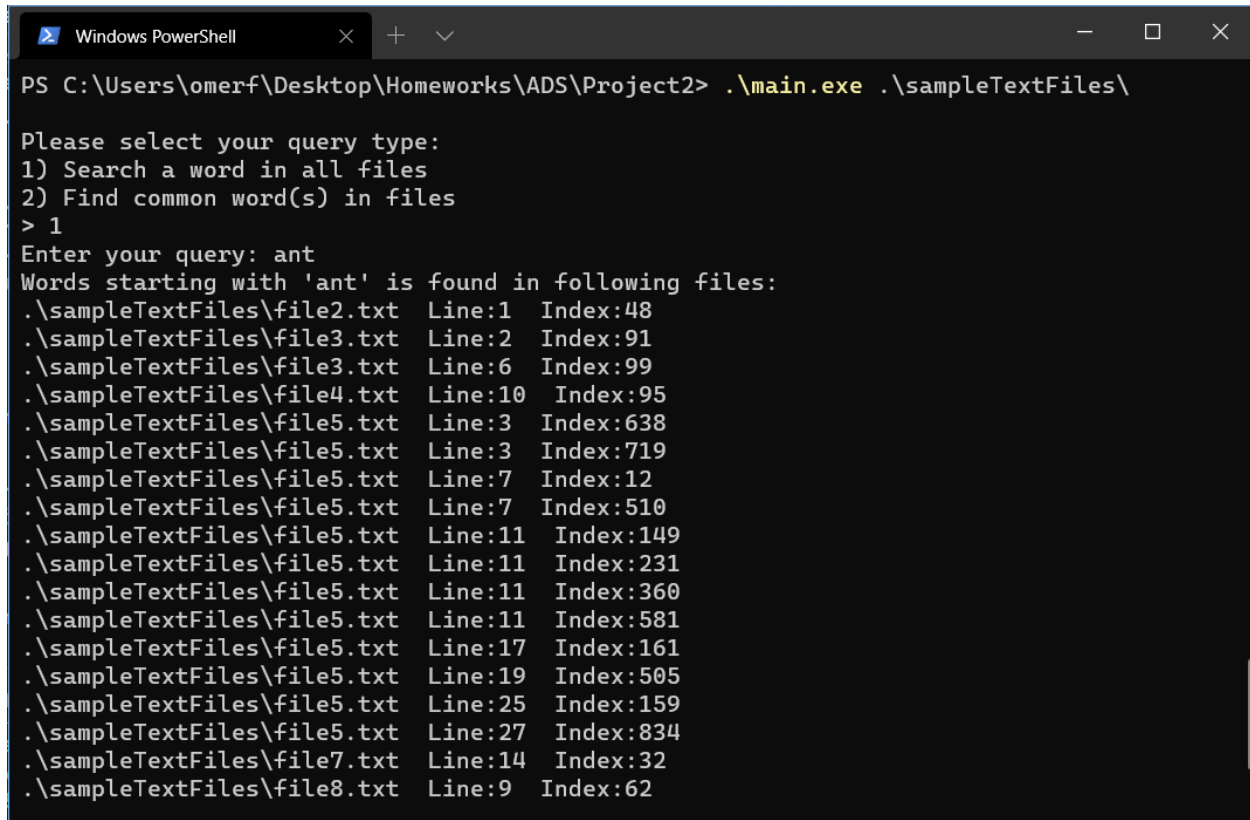
```
g++ main.cpp -o main -std=c++17
```

Executions

Compiled program can be executed as follows:

```
main.exe <folderpath>
```

Sample Executions for 1st



```
Windows PowerShell
PS C:\Users\omerf\Desktop\Homeworks\ADS\Project2> .\main.exe .\sampleTextFiles\

Please select your query type:
1) Search a word in all files
2) Find common word(s) in files
> 1
Enter your query: ant
Words starting with 'ant' is found in following files:
.\sampleTextFiles\file2.txt Line:1 Index:48
.\sampleTextFiles\file3.txt Line:2 Index:91
.\sampleTextFiles\file3.txt Line:6 Index:99
.\sampleTextFiles\file4.txt Line:10 Index:95
.\sampleTextFiles\file5.txt Line:3 Index:638
.\sampleTextFiles\file5.txt Line:3 Index:719
.\sampleTextFiles\file5.txt Line:7 Index:12
.\sampleTextFiles\file5.txt Line:7 Index:510
.\sampleTextFiles\file5.txt Line:11 Index:149
.\sampleTextFiles\file5.txt Line:11 Index:231
.\sampleTextFiles\file5.txt Line:11 Index:360
.\sampleTextFiles\file5.txt Line:11 Index:581
.\sampleTextFiles\file5.txt Line:17 Index:161
.\sampleTextFiles\file5.txt Line:19 Index:505
.\sampleTextFiles\file5.txt Line:25 Index:159
.\sampleTextFiles\file5.txt Line:27 Index:834
.\sampleTextFiles\file7.txt Line:14 Index:32
.\sampleTextFiles\file8.txt Line:9 Index:62
```

Execution 1: All words starting with “ant”

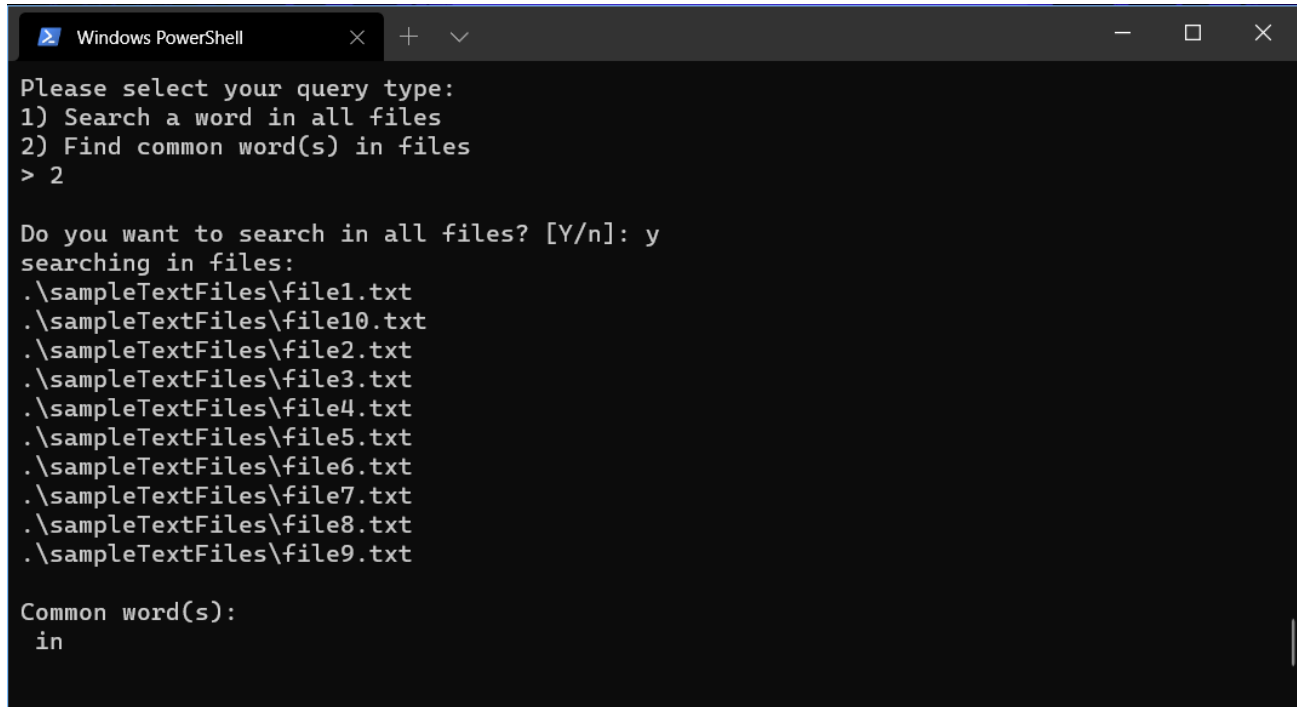
```
Windows PowerShell
Please select your query type:
1) Search a word in all files
2) Find common word(s) in files
> 1
Enter your query: egestas
Words starting with 'egestas' is found in following files:
.\sampleTextFiles\file1.txt Line:4 Index:65
.\sampleTextFiles\file2.txt Line:2 Index:89
.\sampleTextFiles\file3.txt Line:9 Index:34
.\sampleTextFiles\file4.txt Line:9 Index:86
.\sampleTextFiles\file5.txt Line:5 Index:339
.\sampleTextFiles\file5.txt Line:17 Index:413
.\sampleTextFiles\file5.txt Line:17 Index:637
.\sampleTextFiles\file5.txt Line:21 Index:174
.\sampleTextFiles\file5.txt Line:21 Index:251
.\sampleTextFiles\file5.txt Line:23 Index:447
.\sampleTextFiles\file5.txt Line:23 Index:608
.\sampleTextFiles\file6.txt Line:15 Index:49
.\sampleTextFiles\file7.txt Line:2 Index:12
.\sampleTextFiles\file7.txt Line:13 Index:15
.\sampleTextFiles\file7.txt Line:20 Index:5
.\sampleTextFiles\file8.txt Line:3 Index:44
```

Execution 2: All words starting with “egestas”

```
Windows PowerShell
Please select your query type:
1) Search a word in all files
2) Find common word(s) in files
> 1
Enter your query: 11
Words starting with '11' is found in following files:
.\sampleTextFiles\file10.txt Line:1 Index:140
.\sampleTextFiles\file9.txt Line:1 Index:140
.\sampleTextFiles\file10.txt Line:5 Index:6
.\sampleTextFiles\file9.txt Line:5 Index:6
.\sampleTextFiles\file10.txt Line:7 Index:67
.\sampleTextFiles\file9.txt Line:7 Index:67
.\sampleTextFiles\file10.txt Line:9 Index:6
.\sampleTextFiles\file9.txt Line:9 Index:6
.\sampleTextFiles\file10.txt Line:11 Index:65
.\sampleTextFiles\file10.txt Line:13 Index:25
```

Execution 3: All words starting with “11”

Sample Executions for 2nd

A screenshot of a Windows PowerShell window. The title bar shows 'Windows PowerShell' with standard window controls. The terminal text is as follows:

```
Please select your query type:
1) Search a word in all files
2) Find common word(s) in files
> 2

Do you want to search in all files? [Y/n]: y
searching in files:
.\sampleTextFiles\file1.txt
.\sampleTextFiles\file10.txt
.\sampleTextFiles\file2.txt
.\sampleTextFiles\file3.txt
.\sampleTextFiles\file4.txt
.\sampleTextFiles\file5.txt
.\sampleTextFiles\file6.txt
.\sampleTextFiles\file7.txt
.\sampleTextFiles\file8.txt
.\sampleTextFiles\file9.txt

Common word(s):
in
```

Execution 4: Common words in all files in sampleTextFiles directory

```
Windows PowerShell
2) Find common word(s) in files
> 2

Do you want to search in all files? [Y/n]: n

Available files:
1) .\sampleTextFiles\file1.txt
2) .\sampleTextFiles\file10.txt
3) .\sampleTextFiles\file2.txt
4) .\sampleTextFiles\file3.txt
5) .\sampleTextFiles\file4.txt
6) .\sampleTextFiles\file5.txt
7) .\sampleTextFiles\file6.txt
8) .\sampleTextFiles\file7.txt
9) .\sampleTextFiles\file8.txt
10) .\sampleTextFiles\file9.txt
How many files you want to include: 6
Please enter the numbers of the files you want to include (one per line):
> 5
> 2
> 4
> 1
> 6
> 8
searching in files:
.\sampleTextFiles\file4.txt
.\sampleTextFiles\file10.txt
.\sampleTextFiles\file3.txt
.\sampleTextFiles\file1.txt
.\sampleTextFiles\file5.txt
.\sampleTextFiles\file7.txt

Common word(s):
at
in
```

Execution 5: Common words in selected files


```
Windows PowerShell
Please select your query type:
1) Search a word in all files
2) Find common word(s) in files
> 2

Do you want to search in all files? [Y/n]: n

Available files:
1) .\sampleTextFiles\file1.txt
2) .\sampleTextFiles\file10.txt
3) .\sampleTextFiles\file2.txt
4) .\sampleTextFiles\file3.txt
5) .\sampleTextFiles\file4.txt
6) .\sampleTextFiles\file5.txt
7) .\sampleTextFiles\file6.txt
8) .\sampleTextFiles\file7.txt
9) .\sampleTextFiles\file8.txt
10) .\sampleTextFiles\file9.txt
How many files you want to include: 3
Please enter the numbers of the files you want to include (one per line):
> 1
> 3
> 5
searching in files:
.\sampleTextFiles\file1.txt
.\sampleTextFiles\file2.txt
.\sampleTextFiles\file4.txt

Common word(s):
ac
aliquet
dapibus
egestas
erat
et
id
in
ipsum
lacus
lorem
magna
malesuada
maximus
nec
nisl
nulla
phasellus
purus
tempor
ut
vel
vivamus
```

Execution 6: Common words file1.txt, file2.txt, file4.txt