Hacettepe University

Department of Computer Engineering

BBM203 Software Laboratory

Spring 2017

Experiment 4

Subject : *String Search*

Submission Date : *04.05.2017*

Due Date : *20-21.05.2017*

Programming Language: *Java SE - JDK8*

BACKGROUND

String searching algorithms are very important studying area of String algorithms that try to find place or existence of key within larger text. There are known algorithms such as Knuth-Morris-Pratt algorithm, Boyer-Moore algorithms etc. These algorithms are generally based on making a single call. However, indexing may be a better method if searching multiple times in the same text. In other words, different approaches may need to be developed to search on stored data.

The ternary search tree(TST) is a good alternative for getting rid of the cost of linkage for each character in the trie and eliminating the empty links. It is also an important advantage for other languages such as Turkish, as it is also more applicable for non-English languages. TST can be called as “ternary search trie” or “prefix tree” in some resources such as our course book.

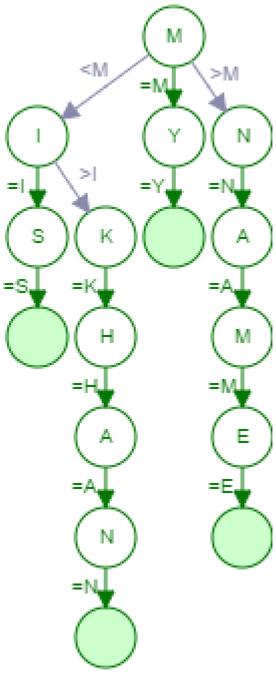

Figure 1 An illustration of TST for text: "MY NAME IS KHAN"

EXPERIMENT

In this experiment, you are aimed to develop a document indexing tool implemented in Java. You will build a Ternary Search Tree and you will traverse it to find files that contains search key. There are two kind of search method; *Phrase* and *Word*. In phrase method, search key must be matched exactly to accept result. In *Word* method, matching any words of search key is accepted. Matching requires perfect match for a word. So if one word is a small part of the other, it will not be enough for this matching. For example, assume that the search term is "abc", it is expected that it does not match the word "abcd". In the same way, "abc def" does not match "abcd ef".

The tool you are going to develop will be able to achieve searching for languages that contains non-English characters such as Turkish letters. It is recommended that you do a preliminary search for the character encoding problem. (Hint: java.util.Locale and UTF-8 can solve many of your problems).

INPUT-OUTPUT FORMAT

The document indexing tool is a command line program and it starts with a single argument which refers to the directory(folder) to index. You will convert all string to lower case according to Turkish Localization (I’s lower case should be ı, not i). After indexing, it asks user a key and search method to operate it. Result will be printed out on the console in given format.

Arguments

<dir>: Path to directory to be indexed. Subdirectories of this directory also have to handled recursively.

Initial output:

<filename#1> indexed

<filename#2> indexed

…

<filename#n> indexed

Commands

search -*type* <key>: It traverses ternary search tree to find matched word/phrase and reports results.

-type: This parameter defines the search method: Word (-w) or Phrase (-W).

<filename>;<matched key>;<begin index>

OR if no result is found;

No result found:<filename>

q

Quit the program

SAMPLE SCENARIO

Assume that, we have three files in a given directory. The names and contents of files are as follows:

File1.txt

Mustafa Kemal Atatürk 19 May 1881 (conventional) – 10 November 1938) was a Turkish army officer, revolutionary, and founder of the Republic of Turkey, serving as its first President from 1923 until his death in 1938.

File2.txt

Atatürk came to prominence for his role in securing the Ottoman Turkish victory at the Battle of Gallipoli during World War I. Following the Empire's defeat and subsequent dissolution, he led the Turkish National Movement, which resisted against the mainland Turkey's partition among the victorious Allied powers. Establishing a provisional government in present day Turkish capital Ankara, he defeated the forces sent by the Allies, thus, emerging victorious from what is later referred to as the Turkish War of Independence. He subsequently proceeded to abolish the Ottoman Empire and proclaimed the foundation of the Turkish Republic in its place.

File3.txt

As the president of the newly formed Turkish Republic, Atatürk initiated a rigorous program of political, economic, and cultural reforms with the ultimate aim of building a modern and secular nation-state. He made primary education free and compulsory, opening thousands of new schools all over the country. Turkish women received equal civil and political rights during Atatürk's presidency ahead of many Western countries. His government also carried out an extensive policy of Turkification trying to create a single, united and largely homogeneous nation. The Turkish Parliament granted him the surname Atatürk in 1934, which means "Father of the Turks", in recognition of the role he played in building the modern Turkish Republic.

Sample results for the above files are given below. Please note that the first letters of the words are reported according to the search key (look at the output difference between 'turkish Republic' and 'Turkish Republic') and the index of the matched result is printed as the last parameter. Also dot(.) character affect the searching same as any other letter (look at the output difference between Republic and Republic.)

java Exp4 <directory_of_files_above>

```
File1.txt indexed
File2.txt indexed
File3.txt indexed
```

search –w Mustafa Kemal Atatürk

```
File1.txt;Mustafa;1
File1.txt;Kemal;2
File1.txt;Atatürk;3
File2.txt;Atatürk;1
File3.txt;Atatürk;10
```

```
File3.txt;Atatürk;90
```

search –W Mustafa Kemal Atatürk

```
File1.txt;Mustafa Kemal Atatürk;1
```

search –W turkish Republic

```
File2.txt;turkish Republic;96
File3.txt;turkish Republic;9
```

search –W Turkish Republic

```
File2.txt;Turkish Republic;96
File3.txt;Turkish Republic;9
```

search –W turkish Republic.

```
File3.txt;turkish Republic.;111
```

search –w turkish Republic.

```
File1.txt;turkish;14
File2.txt;turkish;12
File2.txt;turkish;33
File2.txt;turkish;55
File2.txt;turkish;77
File2.txt;turkish;95
File3.txt;turkish;8
File3.txt;turkish;48
File3.txt;turkish;84
File3.txt;turkish;110
File3.txt;Republic.;111
```

search –w disappointment

```
No result found:disappointment
```

Word delimiters

You must split text into words by using characters as delimiter given below:

New Line(ASCII:10), White Space (ASCII:32), Horizontal Tab (ASCII:9) and Comma(ASCII:44).

NOTES

- Use Eclipse while development
- Use UNDERSTANDABLE names for your variables, functions and classes (please be sure you are obeying name convention).
- Write READABLE SOURCE CODE blocks.
- Use EXPLANATORY COMMENTS in your source codes.
- Don’t miss the deadline.
- CAUTION: Don’t start to thinking about assignment lately. Experiment requires less code but more algorithmic approach.

- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- You can ask your questions through course’s piazza group and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, source codes and reports
- You will submit your work from https://submit.cs.hacettepe.edu.tr with the file hierarchy as below:

```
|--src
        -- Exp4.java
        -- *.java
```

REFERENCES

[1] *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne

[2] ftp://ftp.cs.hacettepe.edu.tr/pub/dersler/genel/FormatForLabReports.doc