CSCI203 ASSIGNMENT1 REPORT

1. A high-level description of the overall solution strategy

My own program **(JAVA)** initially read the file word by word, separating all words by the space with all lowercase keys and punctutaion removing by using functions with parameters: "String lowerCase(String word)" and "removePunctuation(String word). In the next step, I will add each word object to the binary search tree (In-Order Traversal) by comparing word and word, which are adding left node, right node and increase the counter if the words are similar in lexicographical and length. Then I create a BST object in main method and insert root and nodes into BST with lexicographical and length in order. Then, I sort the BST by using quicksort algorithm in decreasing order of counter word in alphabetical if words have the same counter. Finally, I will close the file and display the BST with sorted order.

2. A complexity analysis of your solution with big-O notation and sufficient justification

By using Binary Search Tree as the main Data Structure, traversal the BST into a List and sort them by using QuickSort algorithm (In Java).

(I) Best Case

- BST Insertion: O(nlogn)
- BST Traversal: O(n)
- QuickSort: O(nlogn)
- **Total: O(nlogn)**

(II) Average Case

- BST Insertion: O(nlogn)
- BST Traversal: O(n)
- QuickSort: O(nlogn)
- **Total: O(nlogn)**

(III) Worst Case

- BST Insertion: O(n^2)
- BST Traversal: O(n)
- QuickSort: O(n^2)
- **Total: O(n^2)**

=> Base on the analysis of DSA combination, I estimated the efficient of my program in average performance.

3. A list of all of the data structures used, and the reasons for use them.

There are the list of DSA are used in my program (JAVA)

(-) Data Structures:

a. Binary Search Tree: Implementing the node (Node Class) which show the data and counter, and position. Then, nodes will be added into the BST by comparing string of each word.

b. List: <ArrayList> is imported to the program by traversal the BST into the list and sorted them words.

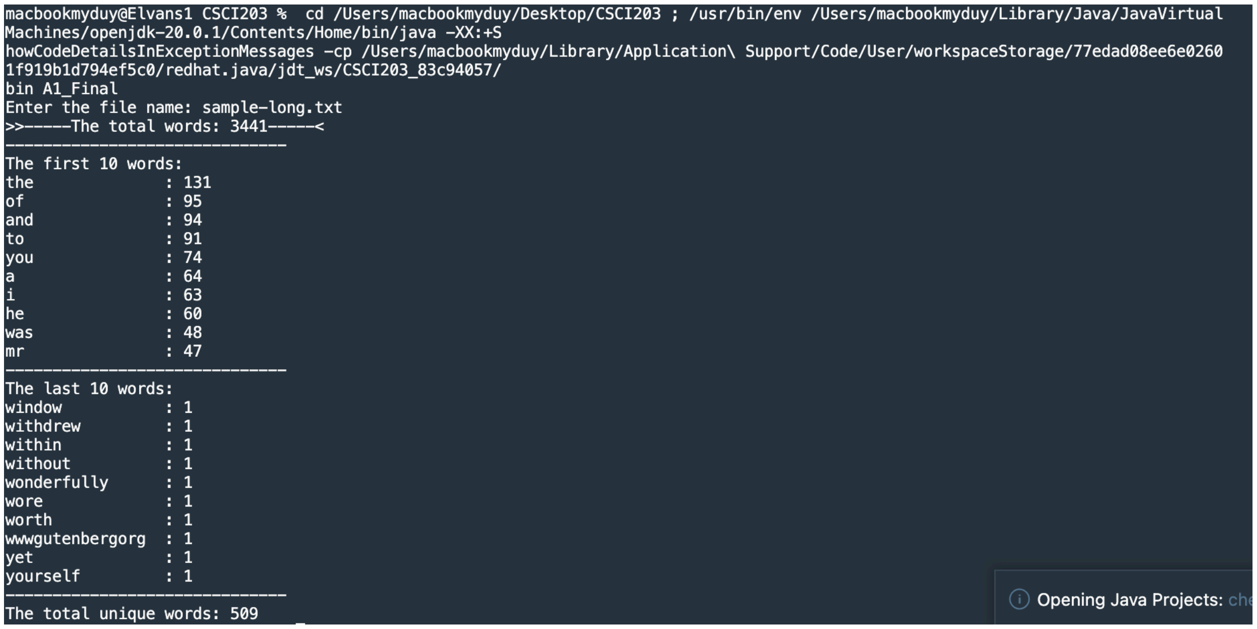
(-) Algorithms:

a. Traversal Tree: collecting all nodes from BST to the list

b. Quick Sort: The List (collecting from BST) will be sorted by using the quick sort algorithm base on the counter of each word in decreasing order (count, alphabet).

=> These DSA because we are going to have the given text file with the small basic data. Based on the requirement of the assignment, BST can be implemented to the program to collect each word as a node straightforwardly. And then, using quick sort algorithm to solve the sorting are very appropriate in question.

4. A snapshot of the compilation and the execution of your program on the provided "sample-long.txt" file.



```
macbookmyduy@Elvans1 CSCI203 %  cd /Users/macbookmyduy/Desktop/CSCI203 ; /usr/bin/env /Users/macbookmyduy/Library/Java/JavaVirtual
Machines/openjdk-20.0.1/Contents/Home/bin/java -XX:+S
howCodeDetailsInExceptionMessages -cp /Users/macbookmyduy/Library/Application\ Support/Code/User/workspaceStorage/77edad08ee6e0260
1f919b1d794ef5c0/redhat.java/jdt_ws/CSCI203_83c94057/
bin A1_Final
Enter the file name: sample-long.txt
>>-----The total words: 3441-----<
------------------------------
The first 10 words:
the              : 131
of               : 95
and              : 94
to               : 91
you              : 74
a                : 64
i                : 63
he               : 60
was              : 48
mr               : 47
------------------------------
The last 10 words:
window           : 1
withdrew         : 1
within           : 1
without          : 1
wonderfully      : 1
wore             : 1
worth            : 1
wwwgutenbergorg  : 1
yet              : 1
yourself         : 1
------------------------------
The total unique words: 509
```

5. **The output produced by your program on the provided “sample-long.txt” file.**

```
Enter the file name: sample-long.txt
>>-----The total words: 3441-----<
------------------------------
The first 10 words:
the          : 131
of           : 95
and          : 94
to           : 91
you          : 74
a            : 64
i            : 63
he           : 60
was          : 48
mr           : 47
------------------------------
The last 10 words:
window              : 1
withdrew            : 1
within              : 1
without             : 1
wonderfully         : 1
wore                : 1
worth               : 1
wwwgutenbergorg     : 1
yet                 : 1
yourself            : 1
------------------------------
The total unique words: 509
```