University of Cape Town

Computer Science assignment 2

Student details: Awonke Mnotoza | MNTAWO002

AIM OF THE ASSIGNMENT:

The goal of this assignment is to test the performance of the AVL Tree to determine if AVL trees really do balance nodes and provide good performance irrespective of the size of the data.

GIVEN DATASETS:

1. GenericsKB.txt

- Each data item consists of a term (which should be used as the key – assume that there won't be duplicate keys in the dataset), sentence and confidence score.

2. GenericsKB-queries.txt

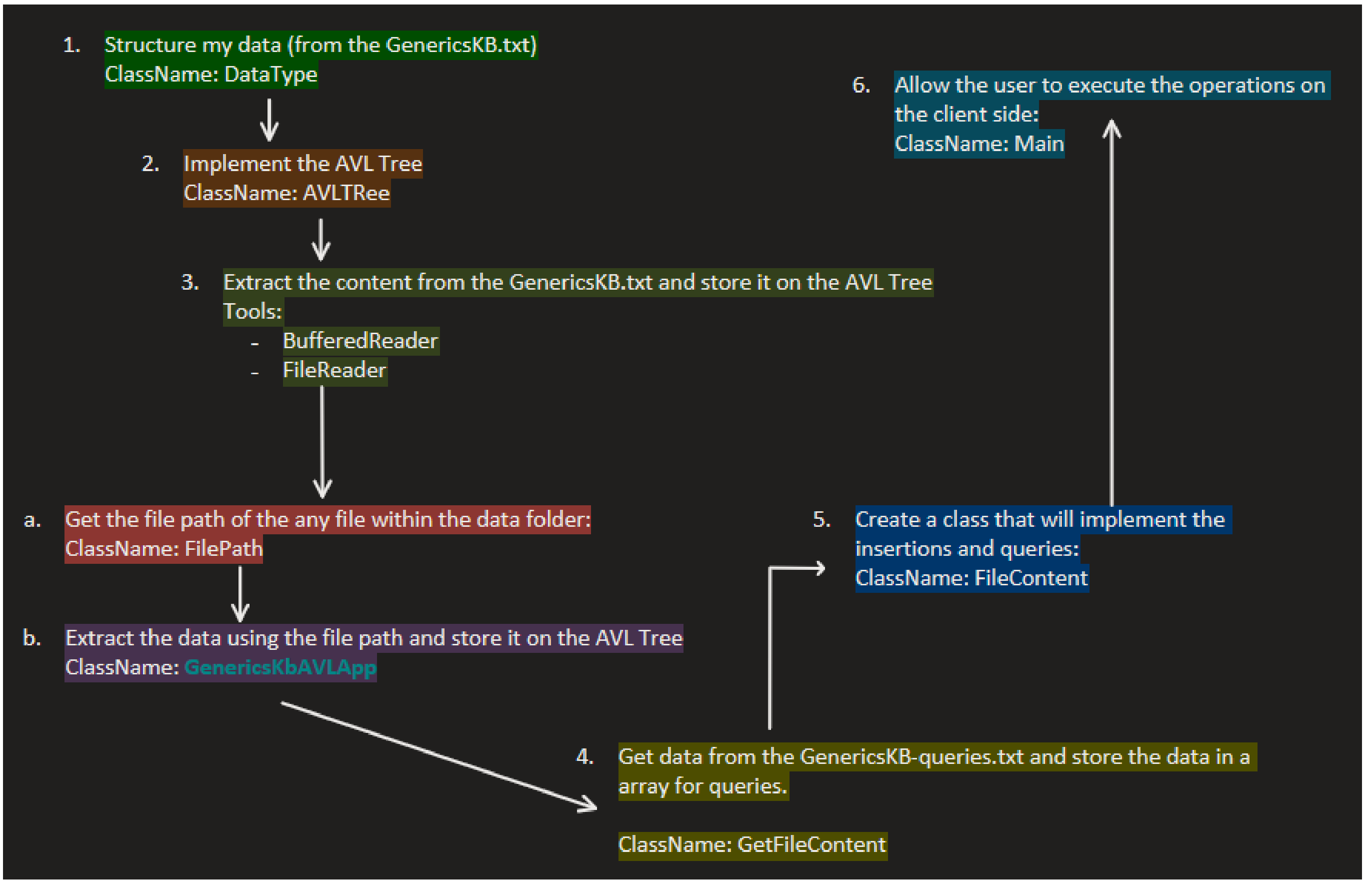
- This file consists of a list of items (one item per line) only.

TASKS TO COMPLETE:

☑ - Create a file called GenericsKbAVLApp that will get the data from the GenericsKB.txt file.
☑ - Store the collected data into an AVL tree.
☑ - Get the data content from the GenericsKB-queries.txt and store the data into an array.
☑ - Perform a search for each of the query items in the array (one at a time) on the AVL Tree.
☑ - Implement instrumentation on the search and insertion methods.
☑ - Report the instrumentation to the user before the project terminates.
☑ - Conduct an experiment for 3 different data sets to demonstrate the best case, average case and the worst case.

OOP DESIGN:

To achieve the tasks above, I need to create a design for my OOP implementation. I made a diagram for this:



1. Structure my data (from the GenericsKB.txt)
ClassName: DataType

2. Implement the AVL Tree
ClassName: AVLTRee

3. Extract the content from the GenericsKB.txt and store it on the AVL Tree
Tools:
- BufferedReader
- FileReader

a. Get the file path of the any file within the data folder:
ClassName: FilePath

b. Extract the data using the file path and store it on the AVL Tree
ClassName: GenericsKbAVLApp

4. Get data from the GenericsKB-queries.txt and store the data in a array for queries.

ClassName: GetFileContent

5. Create a class that will implement the insertions and queries:
ClassName: FileContent

6. Allow the user to execute the operations on the client side:
ClassName: Main

University of Cape Town

Computer Science assignment 2

Student details: Awonke Mnotoza | MNTAWO002

1. Since I know how the data is structured in the GenericsKB.txt, I have created a class that will resemble that data. This class will be used across the whole application for data structuring and extraction.
2. I implemented the AVL from the code we were given on the notes tab on **Amathuba**. I updated it there and there to ensure that it works well with the application. I also gave credits to the creator.
3. Here I wanted to extract the contents from the file GenericsKB.txt or any file that has content with the same structure of the GenericsKB.txt.
    a. To extract the contents, I first need to know where the data is, hence I created this class to get the path of the file.
    b. After getting the file path, I can now get the data within the file. I ensure that the structure of the data is maintained by splitting each line from the file by a tab "\t", then convert the data accordingly.
4. After inserting the data into the AVL Tree, I used to the same approach to get the data from the GenericsKB-queries.txt, or any data that is of similar nature in terms of structure, but I stored the data in a ArrayList so that it will be easy to query the data for search operations on the AVL Tree.
5. After getting all the data, it is now time to perform the executions, I created a java file for this.
6. I created a Main function for the user to interact with the application and be able to perform the operations required to fulfil the aim of this assignment.

THE INSTRUMENTATION

We are required to evaluate the performance of our algorithm. We are told that to achieve this, we need to count the number of executions for each operation (search and insertion).

- I create a Performance file that will achieve this goal.
- For every execution within the insertion and search methods as well as their used classes, I have added a method to increment the count variable.
- Within the Performance class, there is also a method that I will use to evaluate the performance where I will also show the time complexity in a String table format.
- The performance will be shown in the terminal when the user interacts with the application.

THE EXPERIMENT:

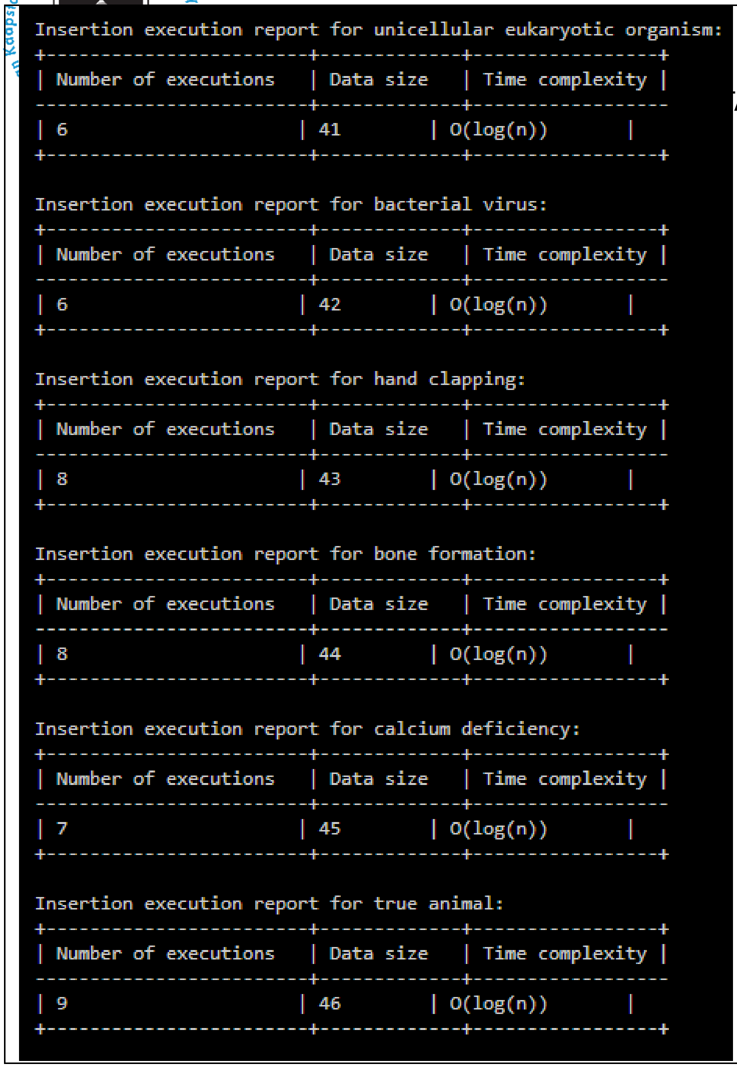
I created 3 sample data files from the GenericsKB.txt and the GenericsKB-queries.txt. I have stored this data on the data under the sample folder.

- The first sample data contains two files which I called Sample50KB.txt and Sample50KB-queries.txt which contains 50 entries each.
- The second sample data contains two files which I called Sample500KB.txt and Sample500KB-queries.txt which contains 500 entries each.
- The third sample data contains two files which I called Sample5000KB.txt and Sample5000KB-queries.txt which contains 5000 entries each.

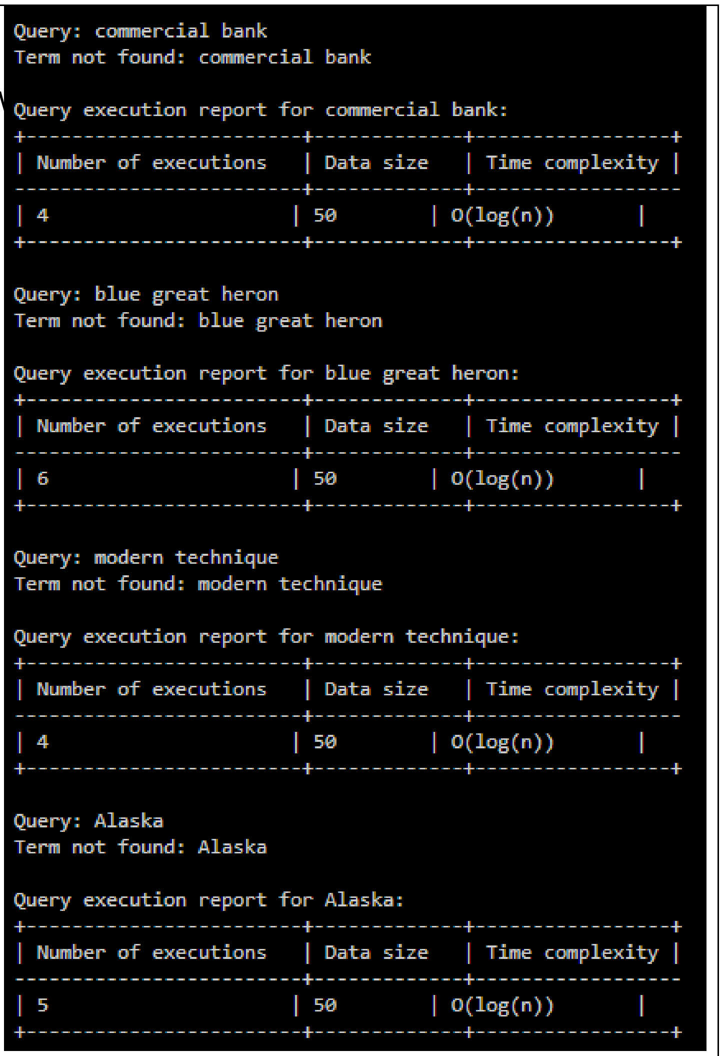
I ran the experiment and got the following results:

SAMPLE 50 (BEST CASE):

Insertion	Search



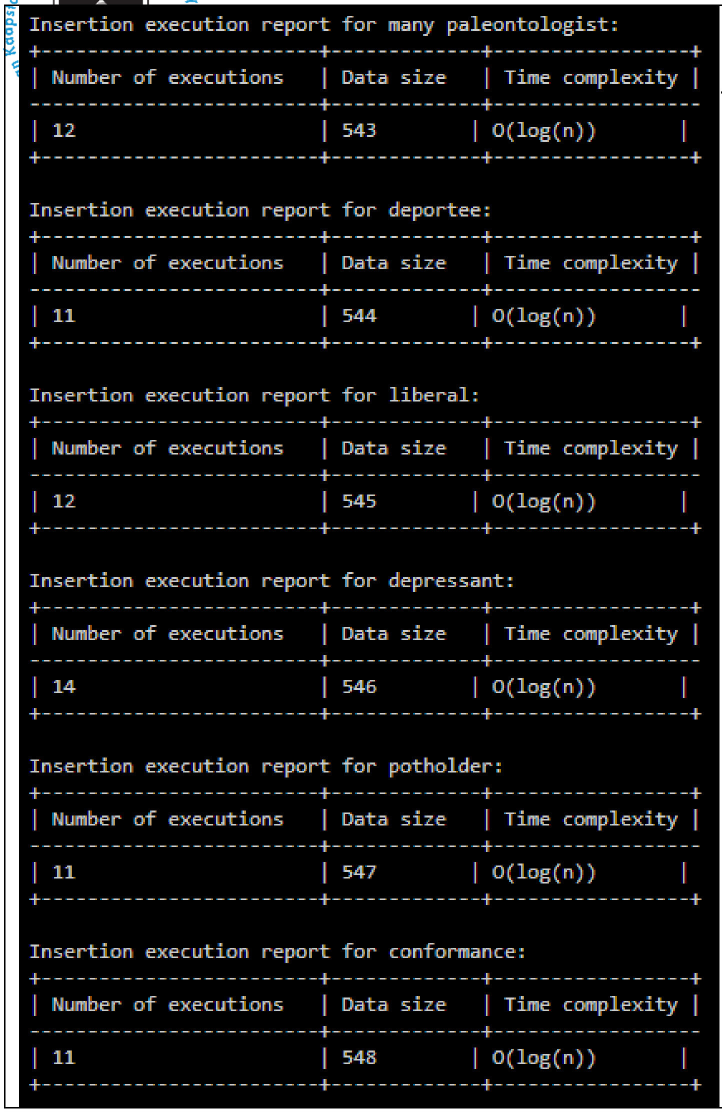
```
Insertion execution report for unicellular eukaryotic organism:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 6                     | 41          | O(log(n))       |
+------------------------+-------------+-----------------+

Insertion execution report for bacterial virus:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 6                     | 42          | O(log(n))       |
+------------------------+-------------+-----------------+

Insertion execution report for hand clapping:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 8                     | 43          | O(log(n))       |
+------------------------+-------------+-----------------+

Insertion execution report for bone formation:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 8                     | 44          | O(log(n))       |
+------------------------+-------------+-----------------+

Insertion execution report for calcium deficiency:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 7                     | 45          | O(log(n))       |
+------------------------+-------------+-----------------+

Insertion execution report for true animal:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 9                     | 46          | O(log(n))       |
+------------------------+-------------+-----------------+
```



```
Query: commercial bank
Term not found: commercial bank

Query execution report for commercial bank:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 4                     | 50          | O(log(n))       |
+------------------------+-------------+-----------------+

Query: blue great heron
Term not found: blue great heron

Query execution report for blue great heron:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 6                     | 50          | O(log(n))       |
+------------------------+-------------+-----------------+

Query: modern technique
Term not found: modern technique

Query execution report for modern technique:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 4                     | 50          | O(log(n))       |
+------------------------+-------------+-----------------+

Query: Alaska
Term not found: Alaska

Query execution report for Alaska:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+------------------
| 5                     | 50          | O(log(n))       |
+------------------------+-------------+-----------------+
```

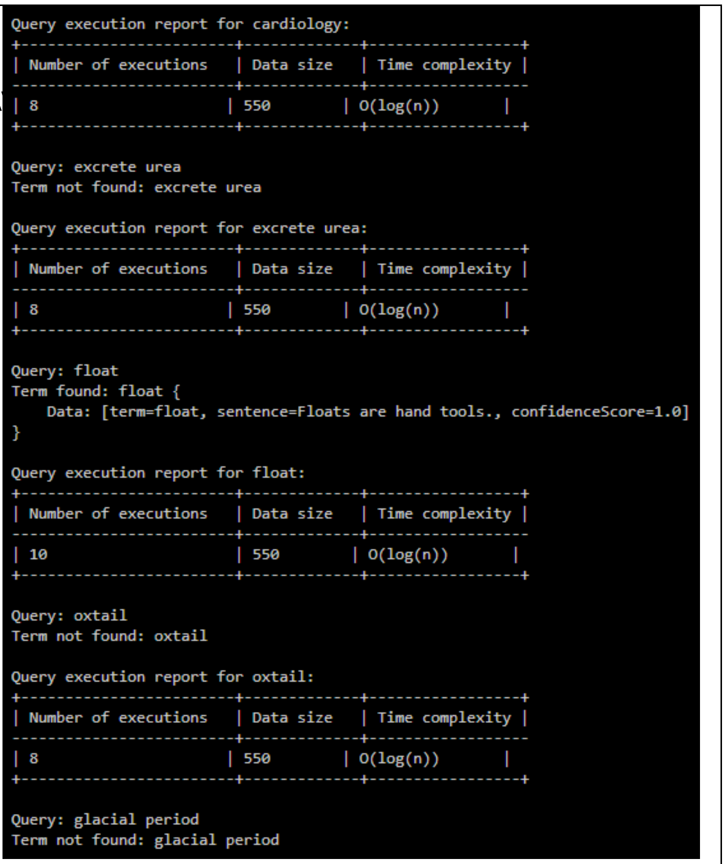
SAMPLE 500 (AVERAGE CASE):

Insertion	Search



```
Insertion execution report for many paleontologist:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 12                     | 543        | O(log(n))        |
+------------------------+-------------+-----------------+

Insertion execution report for deportee:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 11                     | 544        | O(log(n))        |
+------------------------+-------------+-----------------+

Insertion execution report for liberal:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 12                     | 545        | O(log(n))        |
+------------------------+-------------+-----------------+

Insertion execution report for depressant:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 14                     | 546        | O(log(n))        |
+------------------------+-------------+-----------------+

Insertion execution report for potholder:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 11                     | 547        | O(log(n))        |
+------------------------+-------------+-----------------+

Insertion execution report for conformance:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 11                     | 548        | O(log(n))        |
+------------------------+-------------+-----------------+
```

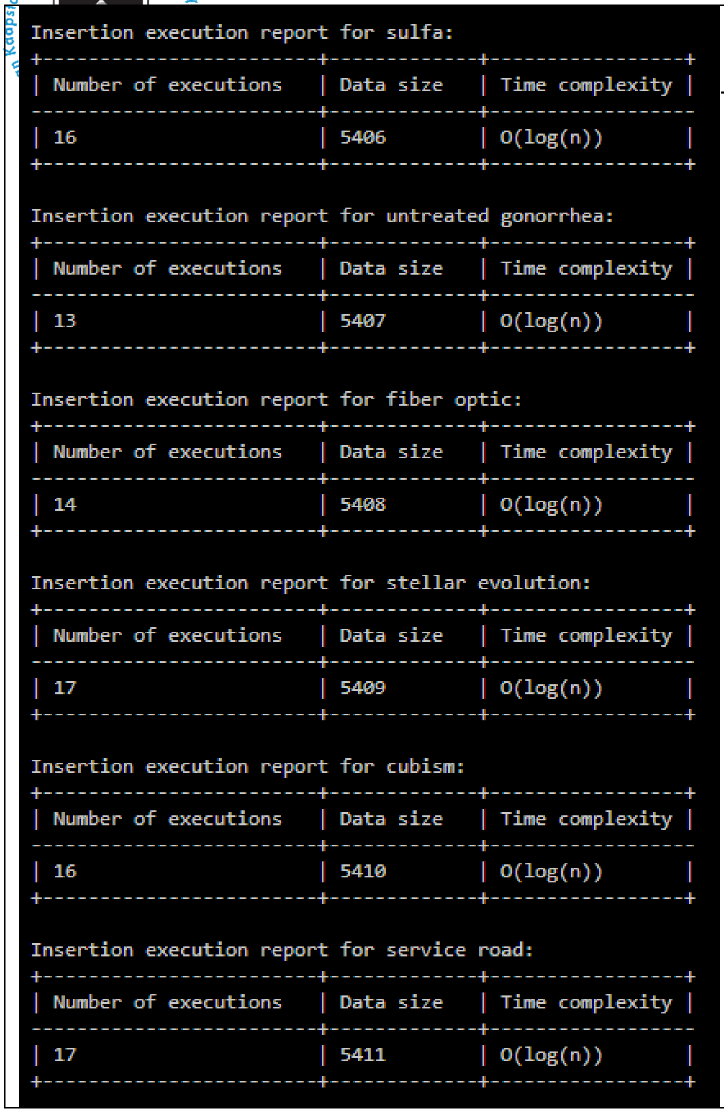
TA



```
Query execution report for cardiology:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 8                     | 550        | O(log(n))        |
+------------------------+-------------+-----------------+

Query: excrete urea
Term not found: excrete urea

Query execution report for excrete urea:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 8                     | 550        | O(log(n))        |
+------------------------+-------------+-----------------+

Query: float
Term found: float {
    Data: [term=float, sentence=Floats are hand tools., confidenceScore=1.0]
}

Query execution report for float:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 10                     | 550        | O(log(n))        |
+------------------------+-------------+-----------------+

Query: oxtail
Term not found: oxtail

Query execution report for oxtail:
+------------------------+-------------+-----------------+
| Number of executions   | Data size   | Time complexity |
-------------------------+-------------+-----------------
| 8                     | 550        | O(log(n))        |
+------------------------+-------------+-----------------+

Query: glacial period
Term not found: glacial period
```

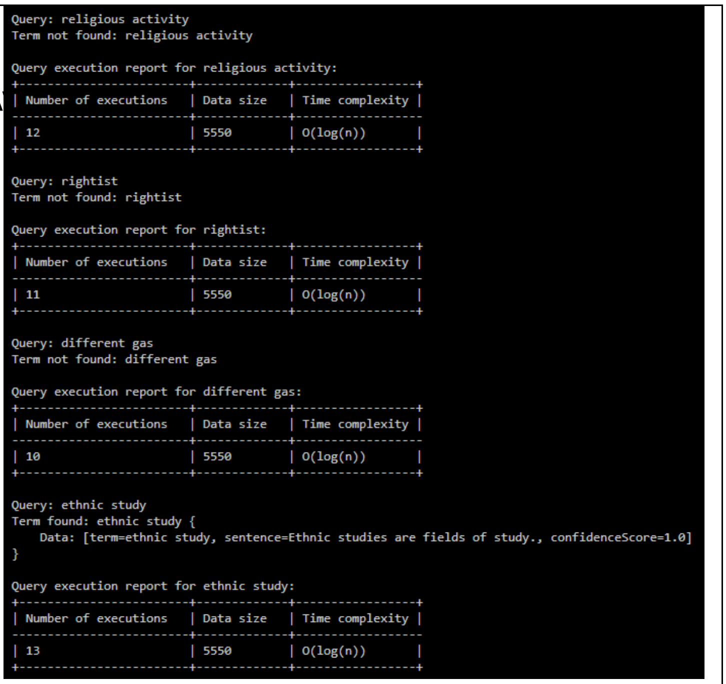
SAMPLE 5000 (WORST CASE):

Insertion	Search



```
Insertion execution report for sulfa:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 16                     | 5406       | O(log(n))       |
+------------------------+------------+-----------------+

Insertion execution report for untreated gonorrhea:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 13                     | 5407       | O(log(n))       |
+------------------------+------------+-----------------+

Insertion execution report for fiber optic:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 14                     | 5408       | O(log(n))       |
+------------------------+------------+-----------------+

Insertion execution report for stellar evolution:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 17                     | 5409       | O(log(n))       |
+------------------------+------------+-----------------+

Insertion execution report for cubism:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 16                     | 5410       | O(log(n))       |
+------------------------+------------+-----------------+

Insertion execution report for service road:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 17                     | 5411       | O(log(n))       |
+------------------------+------------+-----------------+
```

TA



```
Query: religious activity
Term not found: religious activity

Query execution report for religious activity:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 12                     | 5550       | O(log(n))       |
+------------------------+------------+-----------------+

Query: rightist
Term not found: rightist

Query execution report for rightist:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 11                     | 5550       | O(log(n))       |
+------------------------+------------+-----------------+

Query: different gas
Term not found: different gas

Query execution report for different gas:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 10                     | 5550       | O(log(n))       |
+------------------------+------------+-----------------+

Query: ethnic study
Term found: ethnic study {
    Data: [term=ethnic study, sentence=Ethnic studies are fields of study., confidenceScore=1.0]
}

Query execution report for ethnic study:
+------------------------+------------+-----------------+
| Number of executions   | Data size  | Time complexity |
-------------------------+------------+------------------
| 13                     | 5550       | O(log(n))       |
+------------------------+------------+-----------------+
```

EXPERIMENT ANALYSIS:

- The results came out as I expected them to come out.
- The insertion time complexity and the search time complexity should be the same due to the balancing of the AVL tree for each operation.
- The number of executions nearly remain the same as the data size increases during the insertion stage.
- The number of executions nearly remained the same during the search phase for each operation.
- This means that we have achieved our aim of the assignment.

Description of creativity

- I created an interactive platform where the user can interact with the application.
- To get started, run the following commands on the command line or Powershell. Make sure you have ‘make’ installed in your system.

Command	Purpose
make run_main	Compile all files and run the main method
make run_experiment	Compile all files and runs the experimental method
make generate_all_docs	Generate Javadoc for all files

University of Cape Town

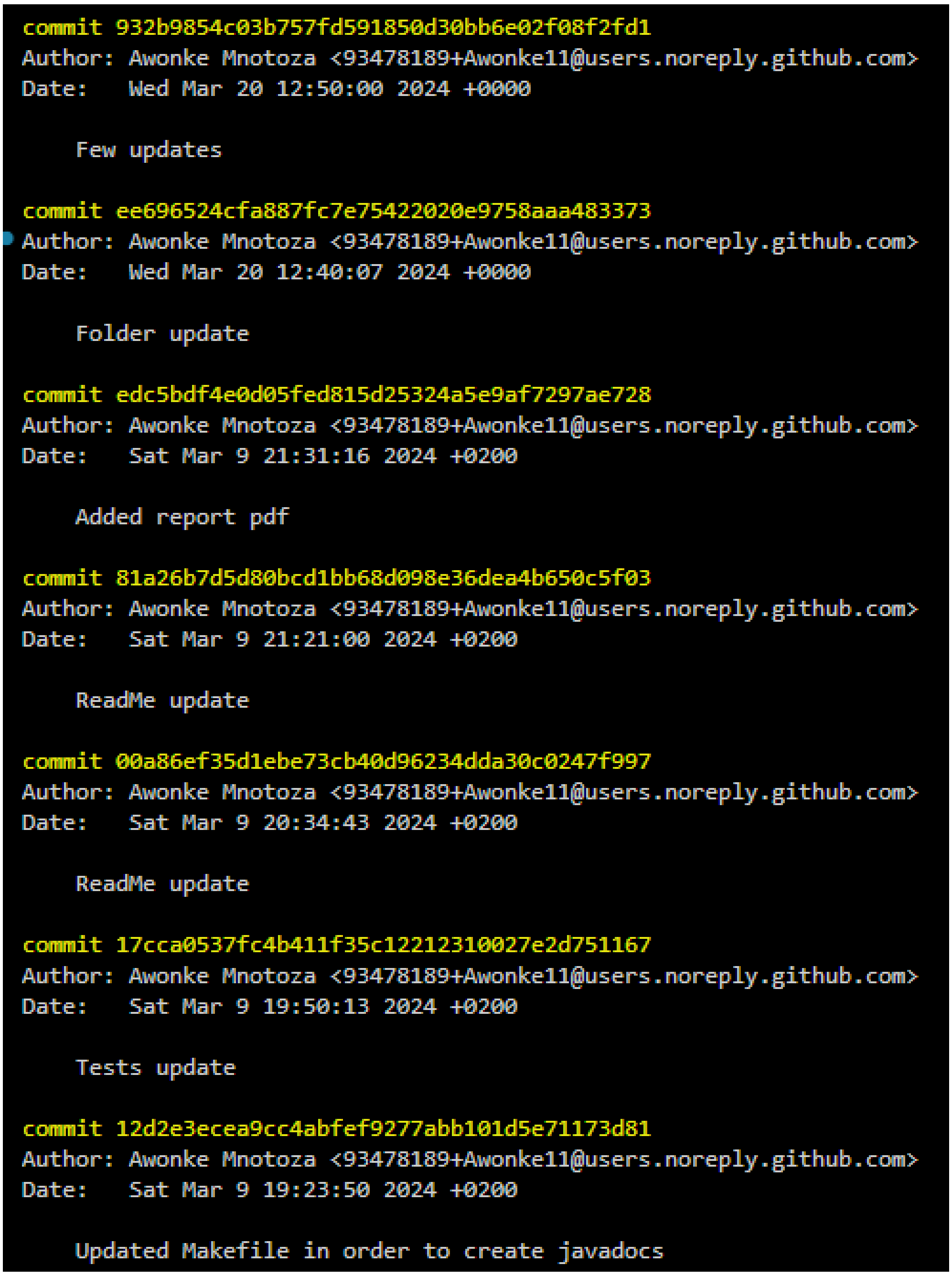
Computer Science assignment 2

Student details: Awonke Mnotoza | MNTAWO002

CONCLUSION

I managed to achieve the aim of the experiment through accessing the performance of insertion and search of the AVL Tree data structure. I can conclude that the performance for the AVL Tree remains the same for the best case, average case, and worst case. This is made possible by the balancing of the AVL Tree, making sure that the min height and max height is of $O(\log n)$ for every execution.

GIT LOG:



```
commit 932b9854c03b757fd591850d30bb6e02f08f2fd1
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Wed Mar 20 12:50:00 2024 +0000

    Few updates

commit ee696524cfa887fc7e75422020e9758aaa483373
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Wed Mar 20 12:40:07 2024 +0000

    Folder update

commit edc5bdf4e0d05fed815d25324a5e9af7297ae728
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Sat Mar 9 21:31:16 2024 +0200

    Added report pdf

commit 81a26b7d5d80bcd1bb68d098e36dea4b650c5f03
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Sat Mar 9 21:21:00 2024 +0200

    ReadMe update

commit 00a86ef35d1ebe73cb40d96234dda30c0247f997
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Sat Mar 9 20:34:43 2024 +0200

    ReadMe update

commit 17cca0537fc4b411f35c12212310027e2d751167
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Sat Mar 9 19:50:13 2024 +0200

    Tests update

commit 12d2e3ecea9cc4abfef9277abb101d5e71173d81
Author: Awonke Mnotoza <93478189+Awonke11@users.noreply.github.com>
Date:   Sat Mar 9 19:23:50 2024 +0200

    Updated Makefile in order to create javadocs
```

GIT REPOSITORY:

uct/assignments/data_structures/assignment_2 at master · Awonke11/uct (github.com)