



UNIVERSITY OF CAPE TOWN

İYUNİVESİTHİ YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

ASSIGNMENT 2 (DS2)

REPORT

Name: Lindokuhle Mdlalose

Stu. No.: MDLLIN028

Date: 22 MARCH 2024

What is my OO Design?

My project comprises of four classes which work together to perform the functionality of my GenericsKB AVL App. This section of the report focusses on what the different classes do and how they work together to perform the functionality of the AVL tree for this project.

Classes Created:

The project has the GenericsKbAVLApp class, which is the main classes in the functionality of the whole project if defined using different methods that are defined in this class. This class enables loading knowledge bases from a file, taking the knowledge base read from the file and storing it in an AVL tree, searching through the AVL tree, rebalancing of the AVL tree so it fits the definition of the AVL tree, and other functionalities needed for the AVL tree.

There are also other two classes which are the Data class and Node class. The Data class stored data objects that are made up of a term, sentence and confidence score. The node class stores the objects of the Data class.

We then have the GenericsKbAVLAppTester app which contains the main method. This class enables user to interact with the whole program by choosing the function they want to perform on the app, and then performing it through this class.

Significance and Interactions:

A standard AVL tree stores elements in form of nodes and then allows for different functions to be performed in the nodes. In the case of this assignment, the GenericsKbAVLApp tree will stores its elements as nodes which are an object of the Node class. The nodes store the data objects which are of the Data class. So, the nodes in the AVL tree created will have the data which is made up of a term, sentence and confidence score.

The GenericsKbAVLAppTester class interacts with the GenericsKbAVLApp class to create the user interface for the program. It allows the user to perform functionalities of the AVL tree by displaying a menu that the user uses to choose the functionality they want to perform and GenericsKbAVLAppTester will communicate with the GenericsKbAVLApp so that functionality can be performed since all the methods of the program are defined in the GenericsKbAVLApp class.

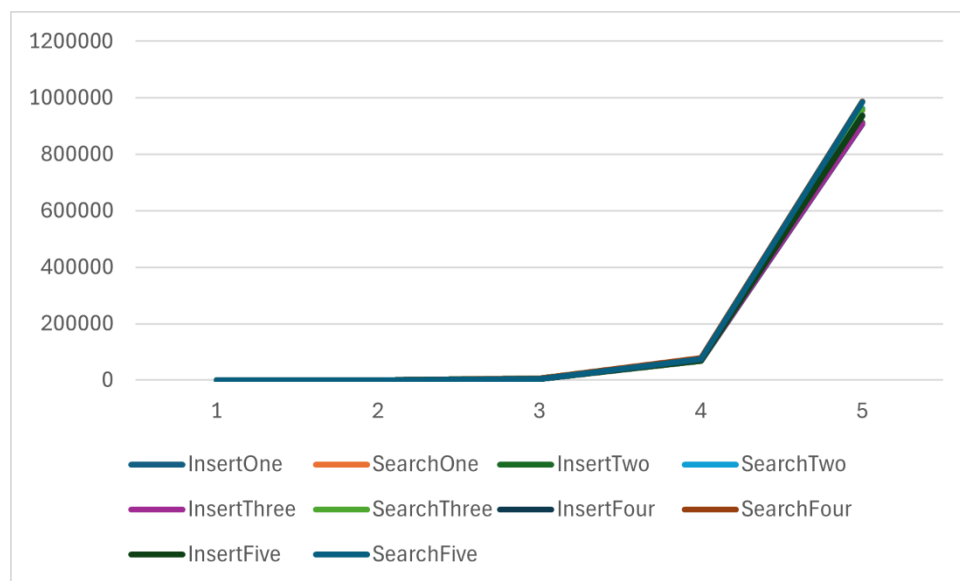
Experiment

For the experiment, I created a method called createShuffledSubsets which I used to shuffle the elements in the knowledge base file, and then when they are shuffled it then allows for the first n elements to be the elements in an array list that will be the subset of the array list that has all the elements. I then created the method called operationCounter which counts the number of insert operations and search operations performed in a subset of size n.

I then ran the operationCounter five times for each subset of size n, and I the recorded the outputs on Excel:

	Trial 1		Trial 2		Trial 3		Trial 4		Trial 5	
Size N:	InsertOne	SearchOne	InsertTwo	SearchTwo	InsertThree	SearchThree	InsertFour	SearchFour	InsertFive	SearchFive
5	10	15	8	13	9	14	10	15	7	12
50	246	296	274	324	271	321	224	274	248	298
500	5815	6315	4831	5331	4863	5363	4863	5363	4289	4789
5000	67236	72236	67232	72232	70679	75679	72314	77314	69272	74272
50000	906848	956848	913348	963348	907778	957778	937718	987718	934911	984911

I then used the table above to create a graph, where the x-axis represents the Size N of the subset (1-5, 2-50, 3-500, 4-5000, 5-50000) and the y-axis represent the number of operations performed by inserting and searching.



I also used the table to find the Best case, Worst case and Average case of inserting and searching:

Insert Operations		Search Operations	
Mininum:	7	Mininum:	12
Maximum:	10	Maximum:	987718
Average:	198932,16	Average:	210043,16

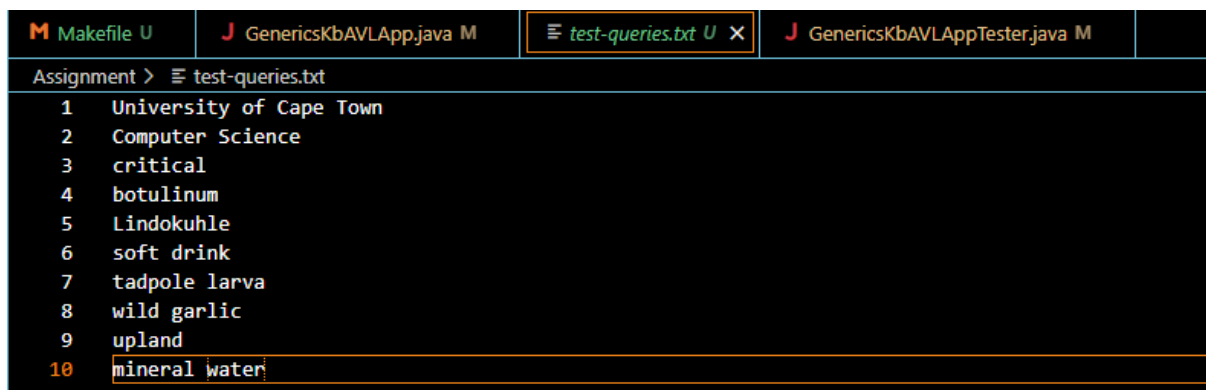
The graph shows that as the size of the AVL tree increases, the number of operations required for inserting a new node or searching for a node also increase.

Testing

For testing, I created a new text file with ten terms, five which are in the GenericsKB.txt file and five which are not in the GenericsKB.txt file.

The first five terms that are not in the GenericsKB.txt file are: University of Cape Town, Computer Science, critical, botulinum and Lindokuhle.

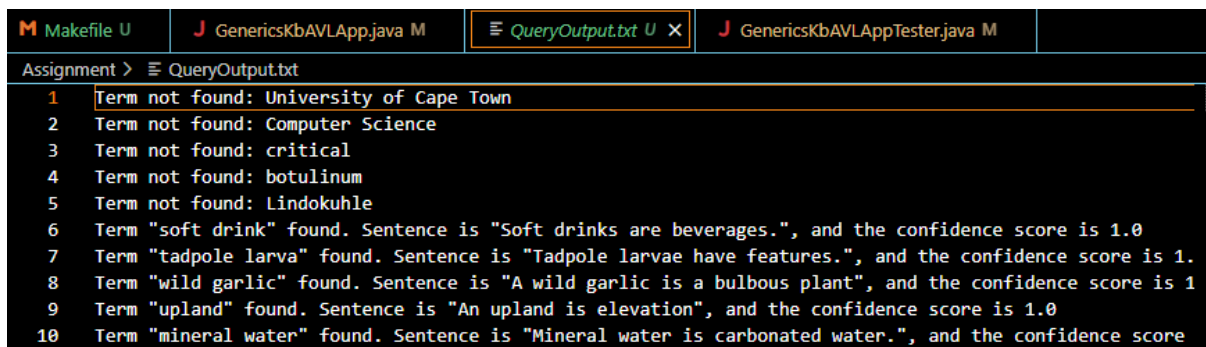
The other five (that are in the GenericsKB.txt file) are: soft drink, tadpole larva, wild garlic, upland and mineral water.



```
M Makefile U  J GenericsKbAVLApp.java M  test-queries.txt U X  J GenericsKbAVLAppTester.java M
Assignment > test-queries.txt
1 University of Cape Town
2 Computer Science
3 critical
4 botulinum
5 Lindokuhle
6 soft drink
7 tadpole larva
8 wild garlic
9 upland
10 mineral water
```

Figure 1: The terms in the query file

When the method that searches queries from the GenericsKB.txt file is executed, it checks if the queries are in the GenericsKB.txt file and will give output of the search in a new file called QueryOutput.txt.



```
M Makefile U  J GenericsKbAVLApp.java M  QueryOutput.txt U X  J GenericsKbAVLAppTester.java M
Assignment > QueryOutput.txt
1 Term not found: University of Cape Town
2 Term not found: Computer Science
3 Term not found: critical
4 Term not found: botulinum
5 Term not found: Lindokuhle
6 Term "soft drink" found. Sentence is "Soft drinks are beverages.", and the confidence score is 1.0
7 Term "tadpole larva" found. Sentence is "Tadpole larvae have features.", and the confidence score is 1.
8 Term "wild garlic" found. Sentence is "A wild garlic is a bulbous plant", and the confidence score is 1
9 Term "upland" found. Sentence is "An upland is elevation", and the confidence score is 1.0
10 Term "mineral water" found. Sentence is "Mineral water is carbonated water.", and the confidence score
```

Figure 2: The output of searching through the GenericsKB.txt using the query file

Creativity

For creativity:

- I created a method that shuffles data in an array list that has the elements of the knowledge base, and then creates a sub list with n number of elements.
- I created the operation counter method to count the number of insert operations and search operations performed in an array list of n elements.
- I created the search method which allows for user to search for a knowledge base in the file using the term.
- I created the countInsertOperations and countSearchOperations methods that evaluated the number of insert and search operations performed.
- I have the displayMenu method which makes it easy for the user to perform functions in the program by selecting what function they want to perform.
- I also used Ascii arts to make the user interface more presentable.

Summary Stats for my use of git:

```
mdllin028@nightmare:~/Documents/Assignment$ git log --oneline | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10;
echo ...; tail -10)
0: a14490e [Made some implementations on my program for the experiment (additional methods and instance variables)]
1: adec616 [Added a new case in the switch loop.]
2: 8717672 [Added the count variables for insert and search. Implemented the incrementation of the count variables in the loadKnowledge
BaseFromFile method and the searchQueriesFromFile method.]
3: 4079cd5 [Removed the main method in the GenericsKbAVLApp class.]
4: 61a70ea [Generated javadoc documentation.]
5: 6543e4e [Added javadoc comments for all methods.]
6: ec74716 [Made some changes in the searchQueriesFromFile method so that the output of the search is saved it a new file, rather than
being printed on screen.]
7: c32b1c6 [Changed the condition of the while loop in the main method.]
8: 23c7d47 [Created the searchQueriesFromFile method.]
9: 440ccfd [Added another case for the queries search method in the switch on the main method (searchQueriesFromFile method not created
yet).]
...
80: 6c4717d [created the searchByTermAndStatement method]
81: 15db5da [Edited the searchByTerm method]
82: 2e385b2 [Created the searchByTerm method]
83: afa04ab [Changed the initial size of the array in the constructor to be equals to the INITIAL_CAPACITY value]
84: e4a5e67 [created the resizeKnowledgeBaseArray size that doubles size of the array if it fills up]
85: abc0638 [added a new static instance variable called INITIAL_CAPACITY]
86: 68893ed [Created the loadKnowledgeBaseFromFile method]
87: 383078e [created the GenericsKBArrayApp class constructor]
88: f02f459 [created instance variables for the GenericsKBArrayApp class]
89: 19e1e25 [Created an inner class called Entry within the GenericsKBArrayApp class]
mdllin028@nightmare:~/Documents/Assignment$
```