



---

# COMPUTER SCIENCE ASSIGNMENT2

---

Anele Chila CHLANE001



# PROBLEM DESCRIPTION

In class we learnt that an AVL Tree is faster than a BST Tree , we are to determine if this is actually true. The problem in hand is to create an Electronic Telephone Directory given a file listing names and details of people, details including address and contact details. . The file list is to be Inserted into a data structure .A data structure is a computational tool that manages data for storing , removing ,replacing and searching for data (other operations for managing data also including). The data structure created in this project is called an AVL Tree. The AVL will be inserted data(details of a person) into, the data can then be managed by operations. The operations we are interested in are the Insert, Search-method and the Delete-method. The methods will accept a query from a query file which will be constructed . The constructed query file will contain twenty names which are to be searched for and deleted from the AVL Tree.

After the AVL Tree has been created and the corresponding operations function accordingly. We will then recall our assignment 1 project which contained a different data structure called a BinarySearchTree(BST) containing operations like search, delete and insert. SearchIt class uses the BST for searching for a query from a query file just like the SearchAVL . We then instrument our code by inserting timekeeping operations , these will help conduct an experiment with SearchIt (from previous assignment) and SearchAVL to demonstrate the speed difference for searching between a BST and an AVL tree. We then measure the speed of insertion, deletion and search operations. We extract names from the data file and use only these names as queries.

# APPLICATION DESIGN

The AVL Tree operations are modified such that they accept two variables. Namely, the full name (key) and the full entry (data) from the data file. The insert operation fills the AVL with the data file . The insert method is **public BinaryTreeNode<String> insert(full name, full entry) in the AVL TREE.**

In my application I have a separate class file for inserting the data file into the AVL . The class file is called **InsertAVL** and it has a method called **dataIn()** which has no parameters. It reads the datafile using a **scanner class** and then inserts the data file into an AVL by calling the insert method in the AVL TREE . It will then return an instance variable containing the data inserted in the AVL.

Now that the data is inserted into the AVL Tree we are now to search for a query from a query file containing a list of 20 words. We first read the query file using **Scanner** again and then take each line as a query and search using the find method from the AVL Tree which returns a

**BinaryTreeNode<String>** . We then store the data into an instance variable then print out the value if it is found , else we print out “**Not found**” if value is not in the AVL Tree.

The **Scanner** class functions in this way: substitute the argument **System.in** (in the **Scanner** constructor) with an appropriate stream that is connected to the text file. The program opens the **Scanner** stream and connects it to a file **QueryFile/data.tar-1** as follows: **Scanner**

**InputStream = null;**

**InputStream = new Scanner( new FileInputStream("QueryFile/data.tar-1"));**

Since the **Scanner** class has no constructor so we need to use another class. In this case, the class **FileInputStream** to convert the file name to an object that can be a suitable argument to the **Scanner** constructor. The methods **nextInt** and **nextLine** read from the text files in exactly the same way as they read from the keyboard(**Scanner** does not only read from the keyboard, the class does not really care about the input source).

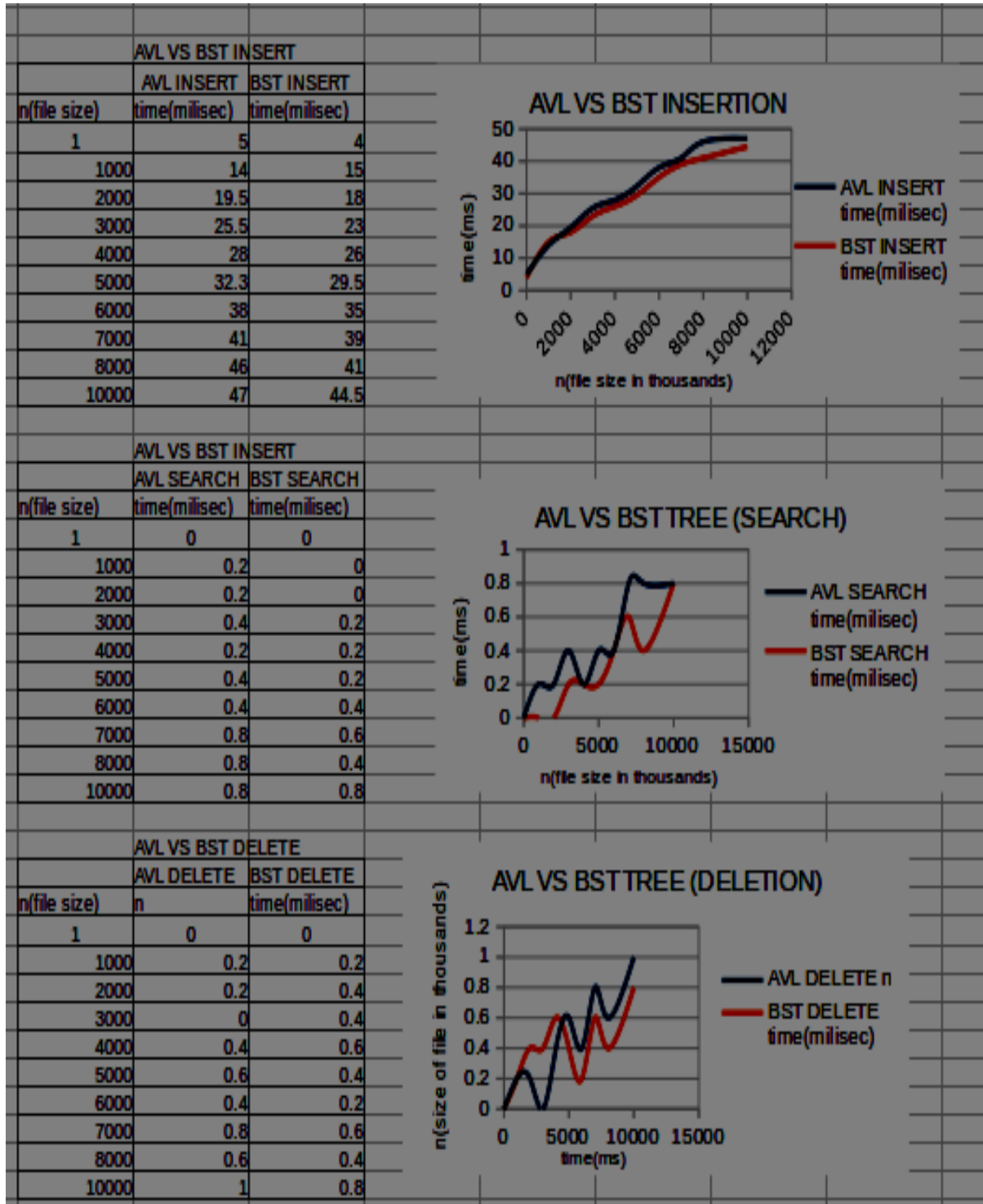
## SearchAVL QueryFile

```
chlane001@sl-dual-152:~/Desktop/CSC2001F/Practical_Two/src$ more QueryFile
Wolf Mariane
Gislason Kenna
Hickle Leone
Moore Gilbert
Eichmann Eliane
Wolff Jaylin
Herzog Ally
Lesch Ephraim
Schaden Vernon
Hessel Pasquale
Eichmann Garry
Pfeffer Kelsie
Stiedemann Johnathon
Casper Jayce
Becker Aurelie
Smithan Janice
Anele Chila
Kuhn Margaret
Mayert Cathy
Kub Heloise
Emmerich Jon
chlane001@sl-dual-152:~/Desktop/CSC2001F/Practical_Two/src$
```

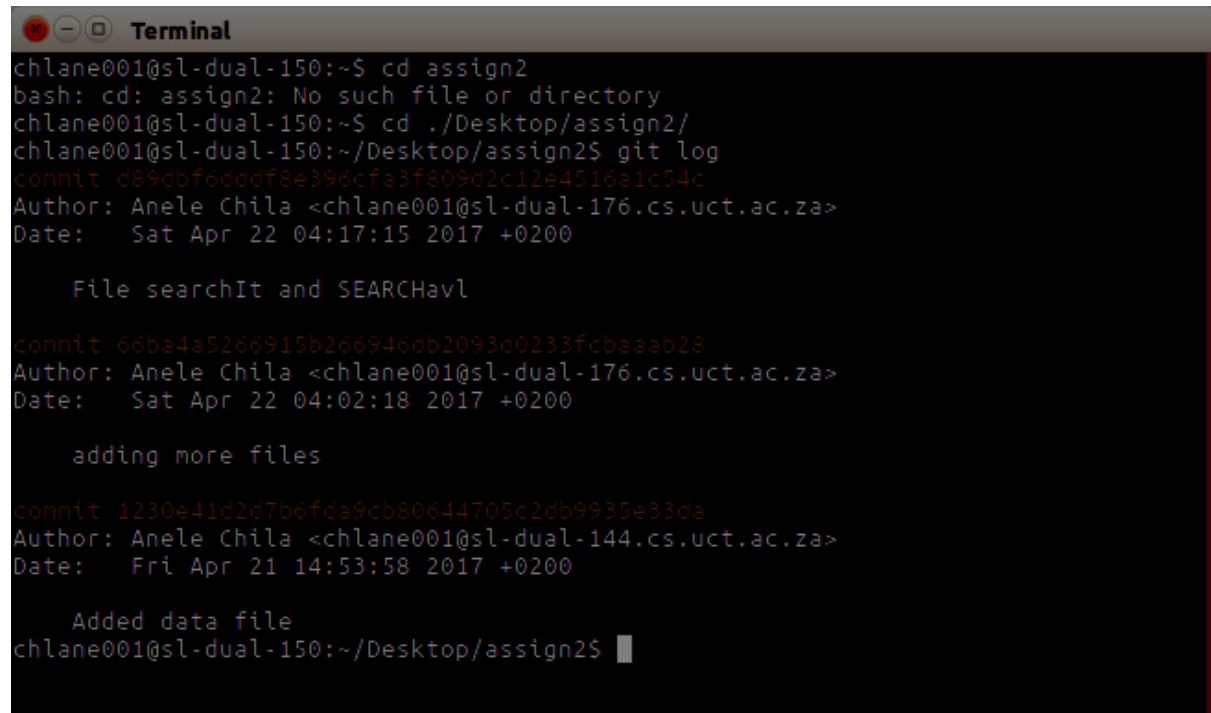
## SearchAVLoutput

```
chlane001@sl-dual-152:~/Desktop/CSC2001F/Practical_Two/src$ more SearchAVLoutput
Time taken of InsertAVL: 47 milliseconds
98289 Alexander Pine #571, Walnut|955.747.0624 x2156|Wolf Mariane
51850 Kianna Squares, Terre Haute|552.531.3674|Gislason Kenna
17386 Stephanie Parks, Palm Springs|018-594-2935 x716|Hickle Leone
97354 Queen Squares, Birmingham|(332)985-4036|Moore Gilbert
89174 Kristy Well, Temple City|720-419-4334|Eichmann Eliane
49784 Schulist Ridge Suite 555, Temecula|583-988-4927 x940|Wolff Jaylin
23005 Morisette Fork Apt. 649, Florence|1-778-083-6571 x13579|Herzog Ally
85380 Robin Freeway, La Habra Heights|(776)957-0613|Lesch Ephraim
31370 Zula Freeway, Jeffersontown|1-417-882-5517 x5439|Schaden Vernon
54637 Kohler Square #222, La Habra|(695)186-8469|Hessel Pasquale
45912 Dwayne Street, Mobile|090-709-3648 x282|Eichmann Garry
36649 Rippin Ports, Mentor|(069)989-8783 x644|Pfeffer Kelsie
30076 King Mews, Hayward|014-934-2377|Stiedemann Johnathon
78637 Florida Cliffs, Blythe|(234)229-3444|Casper Jayce
11608 Candace Court Suite 424, Cerritos|(822)060-1792|Becker Aurelie
23754 Stop R, Anchorage|689.739.7835 x73464|Smithan Janice
Not found
67895 Enard Ferry, Burbank|520.267.1545|Kuhn Margaret
90125 Raven Circle #864, Downey|791-772-8120 x42168|Mayert Cathy
45372 Penthouse, Jasper|(321)417-1788|Kub Heloise
55001 Keeling Inlet #017, Hamilton|480.802.1966 x911|Emmerich Jon
Time taken for SearchAVL : 0 milliseconds
chlane001@sl-dual-152:~/Desktop/CSC2001F/Practical_Two/src$
```

## Tables and Results



## Summary statistics from use of git

A terminal window titled "Terminal" with a dark background and light-colored text. The window shows a series of commands and their outputs. The user attempts to change to a directory named "assign2", which fails because it doesn't exist. They then navigate to the correct path on their desktop and run "git log". The output shows three commit entries, each with a commit hash, a message, the author's name and email, and the date and time. The commit messages are "File searchIt and SEARCHavl", "adding more files", and "Added data file".

```
chlane001@sl-dual-150:~$ cd assign2
bash: cd: assign2: No such file or directory
chlane001@sl-dual-150:~$ cd ./Desktop/assign2/
chlane001@sl-dual-150:~/Desktop/assign2$ git log
commit d89dbf6dddf8e396cfa3f809d2c12e4516a1c54c
Author: Anele Chila <chlane001@sl-dual-176.cs.uct.ac.za>
Date: Sat Apr 22 04:17:15 2017 +0200

    File searchIt and SEARCHavl

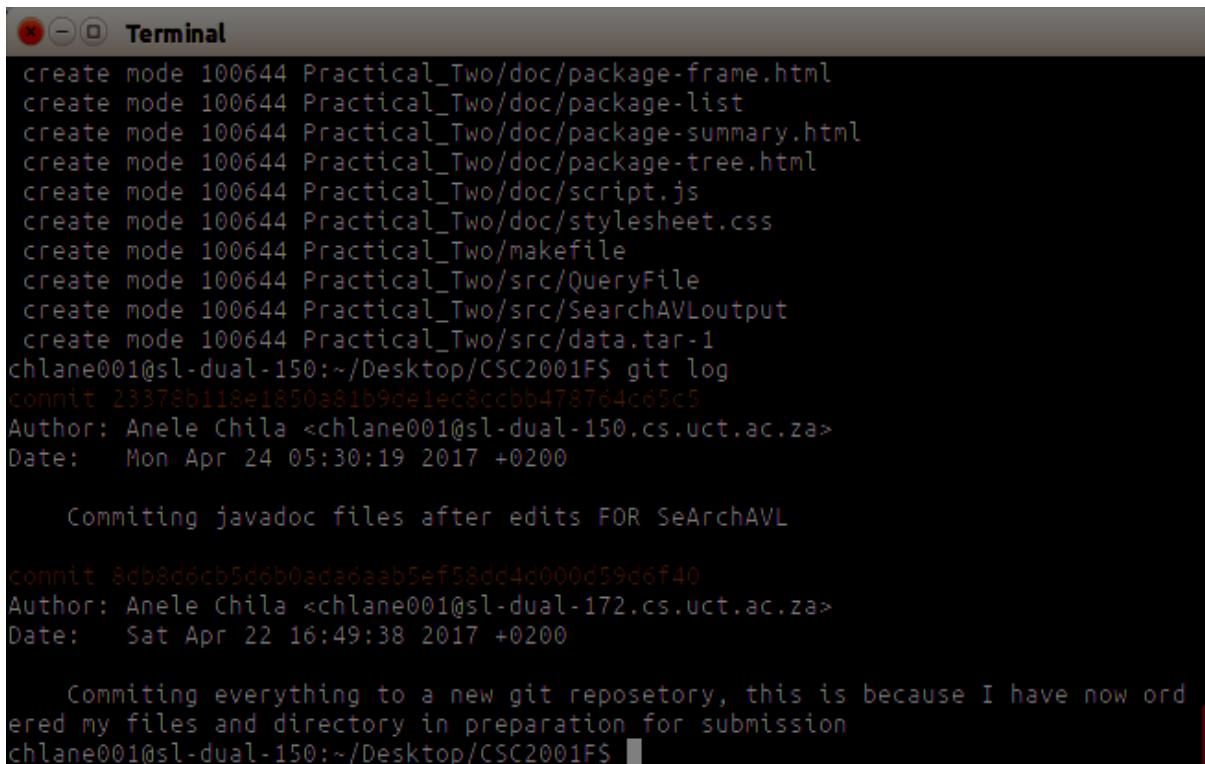
commit 66ba4a5266915b266946db2093d0233fcbaaab28
Author: Anele Chila <chlane001@sl-dual-176.cs.uct.ac.za>
Date: Sat Apr 22 04:02:18 2017 +0200

    adding more files

commit 1230e41d2d7b6fda9cb80644705c2db9935e33da
Author: Anele Chila <chlane001@sl-dual-144.cs.uct.ac.za>
Date: Fri Apr 21 14:53:58 2017 +0200

    Added data file
chlane001@sl-dual-150:~/Desktop/assign2$
```

After putting every file in order

A terminal window titled "Terminal" with a dark background and light-colored text. It shows a series of commands to create files and directories, followed by two git commit operations. The first commit is for "javadoc files after edits FOR SeArchAVL" and the second is for "everything to a new git repository". The terminal output shows the success of these commands and the resulting commit hashes and dates.

```
create mode 100644 Practical_Two/doc/package-frame.html
create mode 100644 Practical_Two/doc/package-list
create mode 100644 Practical_Two/doc/package-summary.html
create mode 100644 Practical_Two/doc/package-tree.html
create mode 100644 Practical_Two/doc/script.js
create mode 100644 Practical_Two/doc/styleSheet.css
create mode 100644 Practical_Two/makefile
create mode 100644 Practical_Two/src/QueryFile
create mode 100644 Practical_Two/src/SearchAVLoutput
create mode 100644 Practical_Two/src/data.tar-1
chlane001@sl-dual-150:~/Desktop/CSC2001FS$ git log
commit 23378b118e1850a81b9de1ec8ccbb478764c65c5
Author: Anele Chila <chlane001@sl-dual-150.cs.uct.ac.za>
Date: Mon Apr 24 05:30:19 2017 +0200

    Committing javadoc files after edits FOR SeArchAVL

commit 8db58d6cb5d6b0ada6aeb5ef58dd4d000d59d6f40
Author: Anele Chila <chlane001@sl-dual-172.cs.uct.ac.za>
Date: Sat Apr 22 16:49:38 2017 +0200

    Committing everything to a new git repository, this is because I have now ordered my files and directory in preparation for submission
chlane001@sl-dual-150:~/Desktop/CSC2001FS$
```

## Discussion

The time it takes to execute an operation (be it insert, delete or find) should be constant for a constant datafile. This is however true only in theory since in practice the computer has other background operations running (processes) in the operating system. The **garbage collector** in java is always running and can sometimes be tight (filled with data) and while java is running code the **collector** may just pop up and clean some space to store current running output. This affects running time. So we have to run multiple times when testing speed for our code and then take the average.

The computer which we are using might also be too fast or too slow to measure accurate readings.

## Conclusion

According to my results, an AVL Tree is SLIGHTLY SLOWER than a BST. This informs me that an AVL is only faster than a BST in the worst case for sorted data. This then tells me that we are working with the average case in our experiment.