
CSC2001F: Assignment 1 report 2021



April 07

Authored by: Kevin Chiloane CHLKEV001

Binary Search Trees

Data structures

The task of the assignment requires a written code to read a text file containing student information(student identity, student name and last name) and store two entries(student identity and student fullname) such as:

CHLKEV001 Kevin Chiloane.

Part 1 : Programs

Created two application `AccessArrayApp` and `AcessBSTApp` to store and retrieve data from text file `oklist`, each application contains two definite methods namely, `printAllStudent` which outputs every entry of inside data and `printStudent` which outputs entry by specific key.

`AccessArrayApp`

Contain class `AccessArrayApp`.

- Which has `ArrayList oklist` to store `oklist` entries.

- 4 methods:

 - `readAndstore` which contains the scanner to iterate over `oklist` and store the entries.

 - main method to create object of class and invoke methods .

 - `printAllStudent` which output every data entry of the `arrayList oklist` from the text file `oklist`.

 - `printStudent` takes in a key which is the identity of the student as type `String` and output a specific data entry by comparing all keys in the `arraylist oklist` and output the match as student student Full name.

`AccessBSTApp`

Contains two classess the main class `AccessBSTApp` and `Node`

`AccessBSTApp` class

-
- Which includes six methods:
 - writeToile which use the FileWriter object to write the number of count operations.
 - countFile creates file to store the results of the operations counted.
 - printAllstudent which output every data entry of the Node currentNode from the text file oklist by an in order traversal.
 - printStudent takes in a key which is the identity of the student identity as type String and output a specific data entry by student identity and student Full name.
 - main to read the text file to the Node object, and create object of class AccessBSTApp and invoke methods .

Class Node

Defines a constructor of Node to take student Id and name.

A to string method to printout desired outputs.

Part 2 : Experiment and testing

The main objective of the assignment was to compare two data structures, linear search and binary search and conclude which one is efficient judged by the speed differences, both Apps, AccessArrayApp representing linear search and AceessBSTApp representing binary search contains variables inside the source codes to count the number of operation counts and storing the results.

Testing each application with 3 known parameters that work and 3 invalid parameters and without any parameters.

Following are snippets of the code running by testing values :

-AccessArrayApp

First five for printAllStudent

```
Activities Terminal Apr 6 13:43
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
File AccessArrayApp.java saved
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1/src$ cd ..
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ make
/usr/bin/javac -d bin/ -cp bin src/AccessArrayApp.java
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp
MLLNOA014 Noah Maluleke
WTBJAY001 Jayden Witbooi
KHZOMA010 Omatla Khoza
MLTLUK019 Luke Malatji
NKNTHA021 Thato Nkuna
WTBOFE020 Ofentse Witbooi
TSHLES016 Lesedi Tshabalala
CHKONT018 Onthatile Chauke
BTHAMO046 Amogelang Buthelezi
DMSMEL001 Melokuhle Adams
DLMORA019 Oratile Dlamini
MGLLET011 Lethabo Mogale
KCCAMA020 ...
```

Last five for printAllStudent

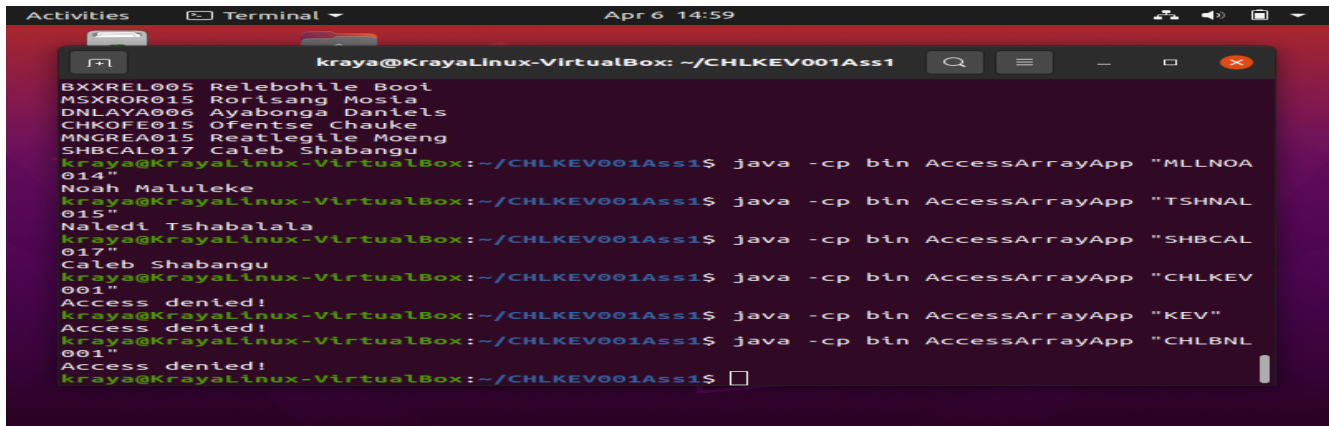
```
Activities Terminal Apr 6 13:43
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
WLLKAM005 Kamohelo Williams
BYSBOK001 Bokamoso Booysen
MTLLUK009 Luke Motloung
TSHTHA025 Thato Tshabalala
WLLTHI003 Iminathi Williams
MNSOMA013 Omatla Mntsi
NGWOF017 Ofentse Ngwenya
MNGENZ020 Enzokuhle Moeng
MDKAMO020 Amohelang Madikizela
DYNONT021 Onthatile Dyantyi
TSHLET016 Letlotlo Tshabalala
BLYRET017 Rethabile Baloyi
MLFOTH024 Othalive Molefe
JCBOMP020 Omphile Jacobs
SHBALU022 Alunamda Shabangu
BTHMIA007 Mia Buthelezi
BXXREL005 Relebohile Booi
MSXROR015 Rorisang Mosia
DNLAYA006 Ayabonga Daniels
CHKOFE015 Ofentse Chauke
MNGREA015 Reatlegile Moeng
SHBCAL017 Caleb Shabangu
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```

printStudent

3 valid inputs

```
Activities Terminal Apr 6 14:58
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
DYNONT021 Onthatile Dyantyi
TSHLET016 Letlotlo Tshabalala
BLYRET017 Rethabile Baloyi
MLFOTH024 Othalive Molefe
JCBOMP020 Omphile Jacobs
SHBALU022 Alunamda Shabangu
BTHMIA007 Mia Buthelezi
BXXREL005 Relebohile Booi
MSXROR015 Rorisang Mosia
DNLAYA006 Ayabonga Daniels
CHKOFE015 Ofentse Chauke
MNGREA015 Reatlegile Moeng
SHBCAL017 Caleb Shabangu
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "MLLNOA
014"
Noah Maluleke
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "TSHNAL
015"
Naledi Tshabalala
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "SHBCAL
017"
Caleb Shabangu
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```

3 invalid inputs

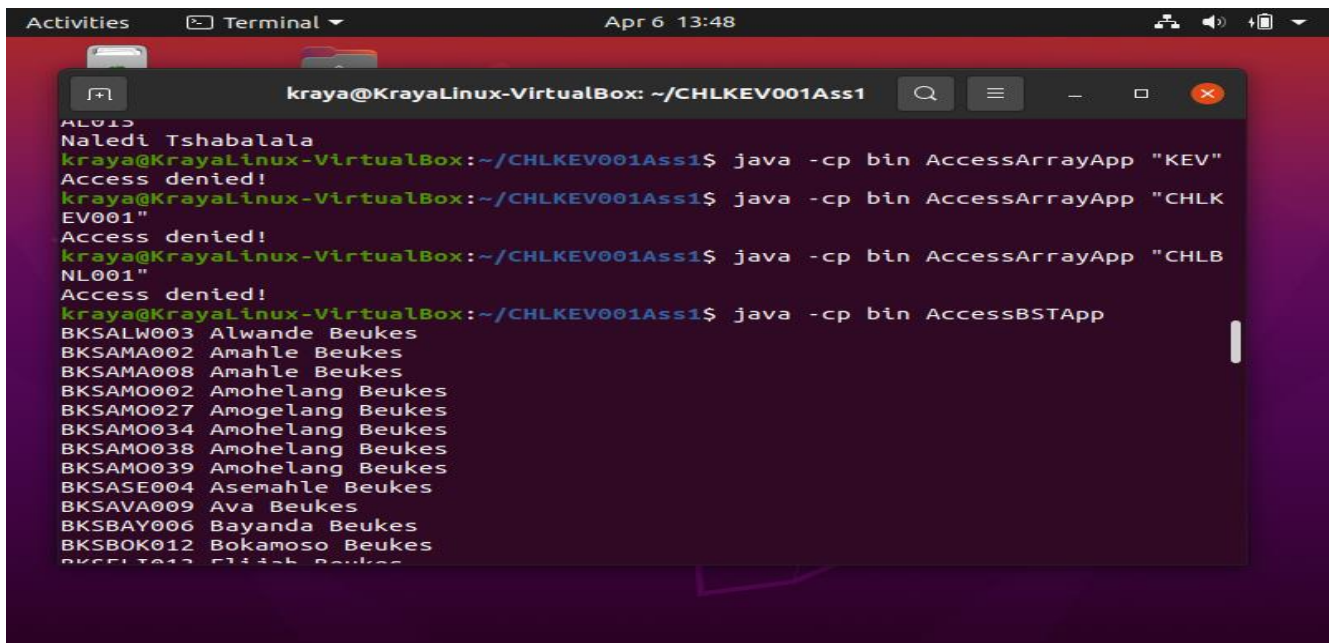


A terminal window titled 'kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1' showing the execution of 'AccessArrayApp'. The program lists a list of students and their IDs. It then prompts for an input. The user enters 'MLLNOA', 'TSHNAL', and 'SHBCAL', which are valid. Then the user enters 'CHLKEV', 'KEV', and 'CHLBNL', which are invalid, resulting in 'Access denied!' messages. The prompt is shown as a box character.

```
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
BXXREL005 Relebohile Boot
MSXROR015 Rorisang Mosta
DNLAYA006 Ayabonga Daniels
CHKOFE015 Ofentse Chauke
MNGREA015 Reatlegile Moeng
SHBCAL017 Caleb Shabangu
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "MLLNOA
014"
Noah Maluleke
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "TSHNAL
015"
Naledi Tshabalala
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "SHBCAL
017"
Caleb Shabangu
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "CHLKEV
001"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "KEV"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "CHLBNL
001"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ 
```

-AccessBSTApp

First five printallStudent



A terminal window titled 'kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1' showing the execution of 'AccessBSTApp'. The program lists a list of students and their IDs. It then prompts for an input. The user enters 'KEV', 'CHLKEV', and 'CHLBNL', which are invalid, resulting in 'Access denied!' messages. Then the user enters 'AccessBSTApp', which is a valid command. The program then prints the first five students from the list.

```
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
AL015
Naledi Tshabalala
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "KEV"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "CHLKEV
001"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessArrayApp "CHLBNL
001"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp
BKSALW003 Alwande Beukes
BKSAMA002 Amahle Beukes
BKSAMA008 Amahle Beukes
BKSAMO002 Amohelang Beukes
BKSAMO027 Amohelang Beukes
BKSAMO034 Amohelang Beukes
BKSAMO038 Amohelang Beukes
BKSAMO039 Amohelang Beukes
BKSASE004 Asemahle Beukes
BKSABA009 Ava Beukes
BKSABY006 Bayanda Beukes
BKSABO012 Bokamoso Beukes
BKSABO013 Bokamoso Beukes
```


Last five printAllStudent

```
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
WTBOFE020 Ofentse Witbooi
WTBOFE037 Ofentse Witbooi
WTBOKU001 Okuhle Witbooi
WTBOLE018 Olerato Witbooi
WTBOMA008 Omaatla Witbooi
WTBOMA017 Omaatla Witbooi
WTBOMP002 Omphile Witbooi
WTBONT011 Onthatile Witbooi
WTBOTH034 Othalive Witbooi
WTBREA002 Reatlegile Witbooi
WTBREA020 Reatlegile Witbooi
WTBRELO11 Relebohile Witbooi
WTBREM005 Remofilwe Witbooi
WTBREN012 Renellwe Witbooi
WTBROR003 Rorisang Witbooi
WTBROR005 Rorisang Witbooi
WTBSIY016 Siyabonga Witbooi
WTBTSH010 Thato Witbooi
WTBTSH002 Tshegofatso Witbooi
WTBTSH025 Tshegofatso Witbooi
WTBTSH028 Tshegofatso Witbooi
WTBWAR001 Warona Witbooi
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```

printStudent

3 valid inputs

```
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
WTBOTH034 Othalive Witbooi
WTBREA002 Reatlegile Witbooi
WTBREA020 Reatlegile Witbooi
WTBRELO11 Relebohile Witbooi
WTBREM005 Remofilwe Witbooi
WTBREN012 Renellwe Witbooi
WTBROR003 Rorisang Witbooi
WTBROR005 Rorisang Witbooi
WTBSIY016 Siyabonga Witbooi
WTBTSH010 Thato Witbooi
WTBTSH002 Tshegofatso Witbooi
WTBTSH025 Tshegofatso Witbooi
WTBTSH028 Tshegofatso Witbooi
WTBWAR001 Warona Witbooi
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "BKSALW00
3"
Alwande Beukes
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "GMDNAL01
6"
Naledi Gumede
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "WTBWAR00
1"
Warona Witbooi
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```

3 invalid inputs

```
kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1
WTBSIY016 Siyabonga Witbooi
WTBTSH010 Thato Witbooi
WTBTSH002 Tshegofatso Witbooi
WTBTSH025 Tshegofatso Witbooi
WTBTSH028 Tshegofatso Witbooi
WTBWAR001 Warona Witbooi
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "BKSALW00
3"
Alwande Beukes
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "GMDNAL01
6"
Naledi Gumede
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "WTBWAR00
1"
Warona Witbooi
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "KEVIN"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "CHLOLV00
1"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ java -cp bin AccessBSTApp "NBNVIN00
1"
Access denied!
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```

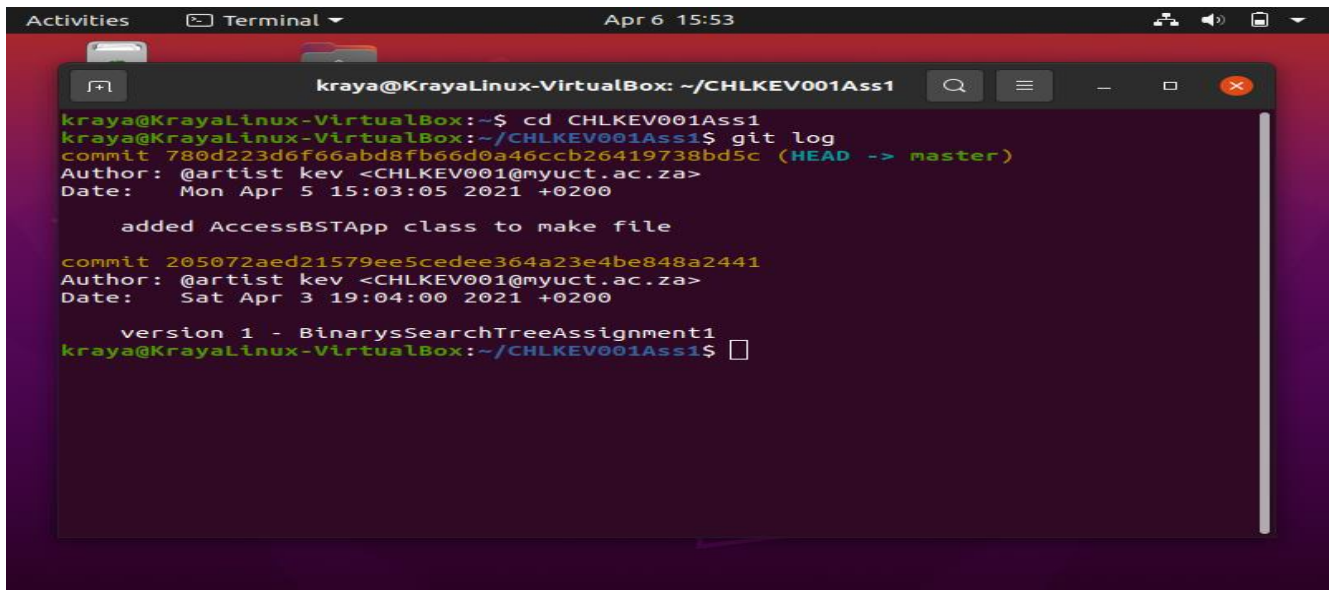
Results

The AccessArrayApp best case $O(1)$ time when the key is at the top of the list, average case $O(n)$ time when the key is at the middle of list and worst $O(n)$ time case when the key is at end of the list.

The AccessBSTApp all operation run in $O(\log n)$ time.

The binary search tree algorithm is more time efficient as it can run all operation with a constant fast time complexity compared to linear search algorithm with it best case slightly slower than the binary search worst case.

Git log

A screenshot of a terminal window titled "kraya@KrayaLinux-VirtualBox: ~/CHLKEV001Ass1". The terminal shows the output of the "git log" command. The first commit is "commit 780d223d6f66abd8fb66d0a46ccb26419738bd5c (HEAD -> master)" by "@artist kev <CHLKEV001@myuct.ac.za>" dated "Mon Apr 5 15:03:05 2021 +0200", with the message "added AccessBSTApp class to make file". The second commit is "commit 205072aed21579ee5cedee364a23e4be848a2441" by the same author dated "Sat Apr 3 19:04:00 2021 +0200", with the message "version 1 - BinarysSearchTreeAssignment1". The prompt "kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1\$" is visible at the bottom.

```
kraya@KrayaLinux-VirtualBox:~$ cd CHLKEV001Ass1
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$ git log
commit 780d223d6f66abd8fb66d0a46ccb26419738bd5c (HEAD -> master)
Author: @artist kev <CHLKEV001@myuct.ac.za>
Date: Mon Apr 5 15:03:05 2021 +0200

    added AccessBSTApp class to make file

commit 205072aed21579ee5cedee364a23e4be848a2441
Author: @artist kev <CHLKEV001@myuct.ac.za>
Date: Sat Apr 3 19:04:00 2021 +0200

    version 1 - BinarysSearchTreeAssignment1
kraya@KrayaLinux-VirtualBox:~/CHLKEV001Ass1$
```