

CSC2001F Assignment 1

RBLCAM001

OO Design

My implementation of OO concepts in this program is as follows: the base data type of the program is an object called Student that consists of a student number and the name of the student that can also be broken down into their name and surname. Therefore, there is an array of student objects which stores all the data points from the oklist textfile as Student objects. The Binary Search Tree also uses this data type to store data from the text file, where each node will contain one student object that each have a student number that they use for their primary key and also then the name attached to that number. I have then created two other classes whose sole purpose is populate the Array/BST using the scanner method. In addition to this I have a class with useful methods called the helper class. This class is used to encapsulate the data from the main class so that the main classes of the different data structures have no control on how the data is implemented or manipulated.

Experiment

The goal of this experiment is to determine the efficiency of the speed of two different data structures: arrays and binary search trees. The experiment uses instrumentation in order to keep track of the number of comparisons that each data structure will do in order to find a specific piece of data. The experiment is done by using a bash file to create subsets of the oklist data by randomly choosing batches of multiples of 500 entries i.e., 500 random entries then 1000 random entries then 1500 and so on. Each time a batch is made in the for loop – each Student number is then sent to the AccessArrayApp/AccessBSTApp where the different data structures will then look through their data and count the number of comparisons they make until they find the corresponding data in the entire list that they are storing. The programs then return the number of comparisons they used in order to find that corresponding piece of data. This data is then used to determine the best case, worst case, and average case of each of the subsets for each of the data structures. This data can then be graphed and used to compare the differences in efficiencies between the data structures.

Trials

AccessArrayApp

Input:

Valid 1: MHLLUB026

Invalid 1: RBLCAM001

Valid 2: MTHRET032

Invalid 2: GEOWAL002

Valid 3: MNGMIN004

Invalid 3: ATTRAE001

```
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayValid1.txt
Lubanzi Mohlala
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayValid2.txt
Rethabile Mthenbu
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayValid3.txt
Minenhle Moeng
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayInvalid1.txt
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayInvalid2.txt
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more ArrayInvalid3.txt
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more head -10 ArrayNoParameters.txt
```

Input:

No parameters

```
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ cat ArrayNoParameters.txt
| head -5
Noah Maluleke MLLNOA014
Jayden Witbooi WTBJAY001
Omaatla Khoza KHZOMA010
Luke Malatji MLTLUK019
Thato Nkuna NKNTHA021
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ cat ArrayNoParameters.txt
| tail -5
Ayabonga Daniels DNLAYA006
Ofentse Chauke CHKOFE015
Reatlegile Moeng MNGREA015
Caleb Shabangu SHBCAL017
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$
```

AccessBSTAPP

Input:

Valid 1: WTBTSH028

Invalid 1: RBLCAM001

Valid 2: NDXRET019

Invalid 2: BNGTAR001

Valid 3: CLTMEL012

Invalid 3: FITMAT001

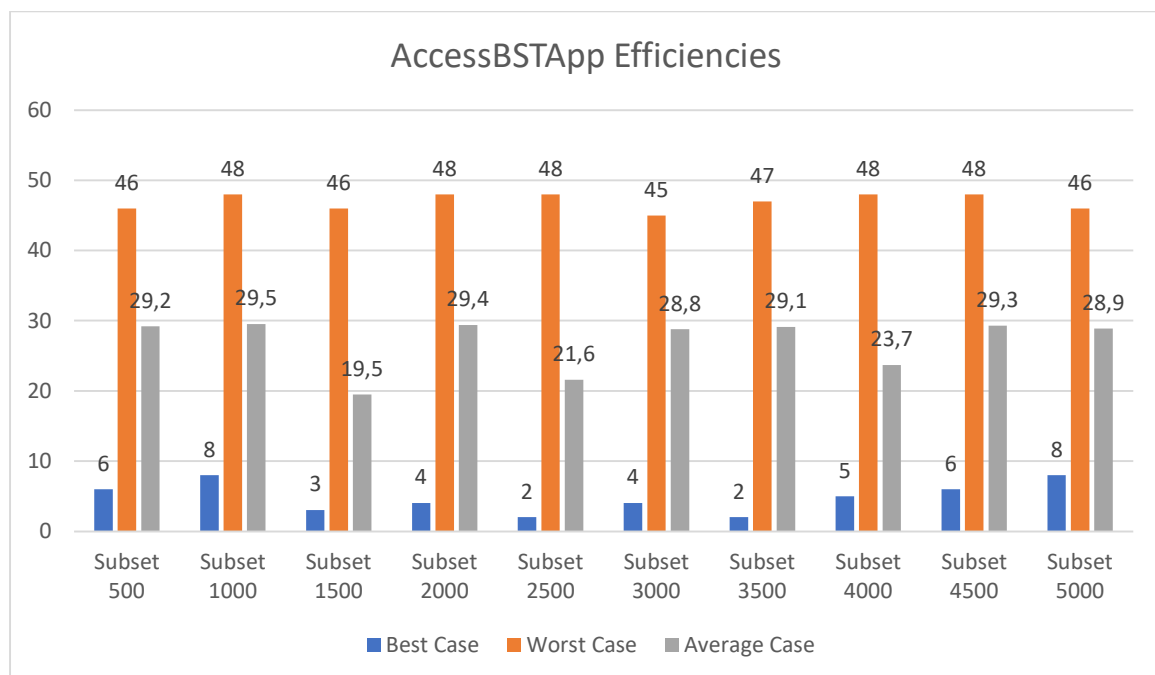
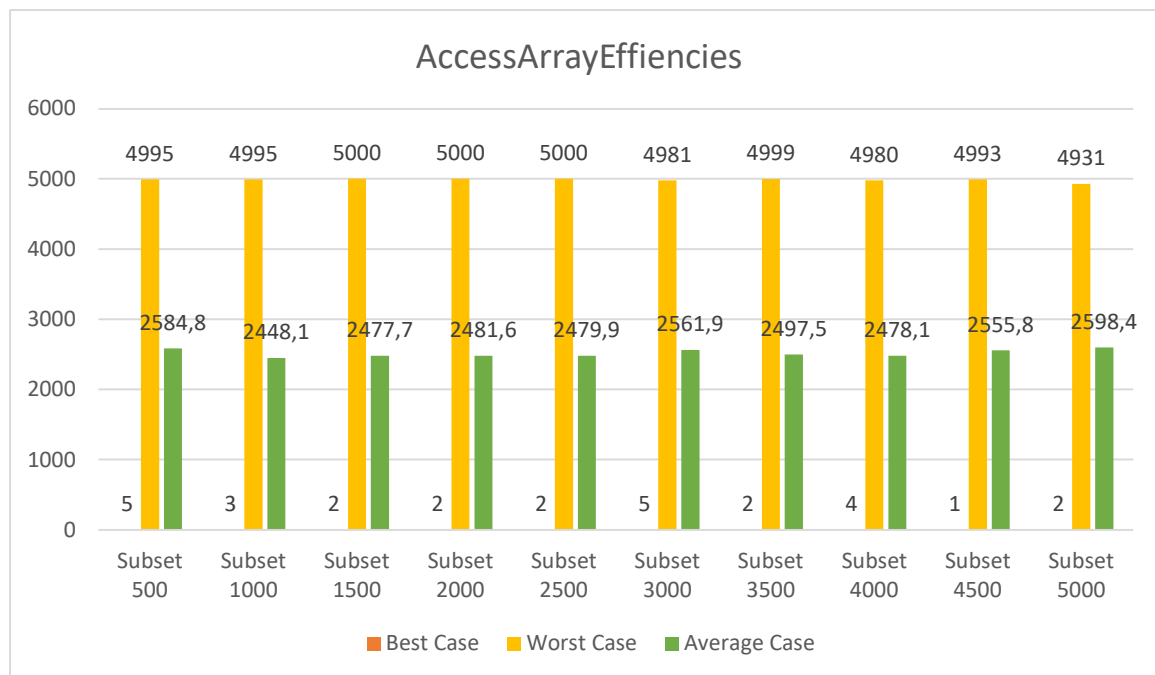
```
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTValid1.txt
Tshegofatso Witbooi
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTValid2.txt
Rethabile Ndou
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTValid3.txt
Melokuhle Cloete
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTInvalid1.txt
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTInvalid2.txt
Comparisons: 16
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ more BSTInvalid3.txt
Comparisons: 13
Access denied!
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$
```

Input:

No parameters

```
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ cat BSTNoParameters.txt
| head -5
Alwande Beukes BKSALW003
Amahle Beukes BKSAMA002
Amahle Beukes BKSAMA008
Amohelang Beukes BKSAM0002
Amogelang Beukes BKSAM0027
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$ cat BSTNoParameters.txt
| tail -5
Thato Witbooi WTBTHA010
Tshegofatso Witbooi WTBTS002
Tshegofatso Witbooi WTBTS025
Tshegofatso Witbooi WTBTS028
Warona Witbooi WTBWAR001
cameron@cameronlaptop-VirtualBox:~/Assignment1/data$
```

Results



From the data presented in the graph, we can see that the data stored in the binary search tree is much more efficient to search through, while the data in the array took an exceptionally long time to process. The array data has an average worst case of about 5000 comparisons and an average average case of about 2500 comparisons. The binary search tree however has an average worst case of only 50 and an average average case of about 30. We can therefore draw the conclusion that the binary search tree is more efficient at searching through data than the array.

Creativity

This assignment was a real challenge for me and required me to do much of my own research in order to get the program to run. The course resources did not cover how to do the experimentation phases and thus much of my time and effort went into figuring that part out. There is a script to test each data type that involves creating the random subsets, parsing them to the program and then also returning the min, max and average cases. The calculations were done with the help of a forum post¹. In my actual data structure programs, I also made use of a special “Helper” class in which methods that I want separated from the main classes can be stored. The methods in here help to make the data structures more encapsulate by taking away the main classes direct access to the data structures.

Git

```
cameron@cameronlaptop-VirtualBox:~/Assignment1$ git log | (ln=0; while read l;
do echo $ln\: $1; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
...
67:
68:
69:
70:
71:
72:
73:
74:
75:
76:

cameron@cameronlaptop-VirtualBox:~/Assignment1$ git log
commit 632c989bad03769a785cad658ca234644b3047d7 (HEAD -> master)
Author: Cameron Rebelo <rblcam001@myuct.ac.za>
Date: Wed Apr 7 18:11:49 2021 +0200

    finished javadoc documentation. project finished

commit de30d408ef63db8d0c77fb1c1393dbb117998ab8
Author: Cameron Rebelo <rblcam001@myuct.ac.za>
Date: Wed Apr 7 17:32:05 2021 +0200

    finished debugging BST test script

commit 11a48131aec4103b26df421cde6015ee36c27971
Author: Cameron Rebelo <rblcam001@myuct.ac.za>
Date: Wed Apr 7 13:19:57 2021 +0200

    Array test script works, still debugging BST test script

commit 037cca26d5421210d092a5d50b7ec4d6c56ea920
Author: Cameron Rebelo <rblcam001@myuct.ac.za>
Date: Tue Apr 6 23:28:31 2021 +0200

    updated files

commit da27bc0a8a620089f1b8ba0d8c0abe23698c4ac4
Author: Cameron Rebelo <rblcam001@myuct.ac.za>
Date: Tue Apr 6 00:40:56 2021 +0200
```

¹ <https://serverfault.com/questions/239496/using-bash-how-can-i-find-out-the-average-max-and-min-from-a-list-of-numbers>

```
added doc folder,  
  
commit 30359b8da7861aaffe1aff1d3b4e384518630f7f3  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Tue Apr 6 00:37:31 2021 +0200  
  
added javadoc comments  
  
commit 0f4ead55e092f38c1f3e73edf9ec9d6d55a38c05  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Mon Apr 5 22:30:22 2021 +0200  
  
Started Array experiment bash file - logging off for session  
  
commit 7d6c4a2138b6706eff698d188f7a586e1f7a6f18  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Mon Apr 5 22:04:55 2021 +0200  
  
Programs finished just before testing phases  
  
commit c52c86d73afcd1749eb941acfd2d3aa7f48a53e9  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Mon Apr 5 13:28:47 2021 +0200  
  
added helper class and StudentArray class for encapsulation purposes  
  
commit b022be6a2f594fa36cfa4158e972461a57505440  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
:
```

```
commit 7a1ee8301ff08cba67d1fde75b7db460b307fd99  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Sat Apr 3 14:43:39 2021 +0200  
  
added BST Code  
  
commit 4e917d679e1d610f1aff29f0f0677fdbcdbfc9aad  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Sat Apr 3 13:38:04 2021 +0200  
  
added student class  
  
commit 97c10bb7347584848cb4b7a0d3ba870cd303a96d  
Author: Cameron Rebelo <rblcam001@myuct.ac.za>  
Date: Sat Apr 3 12:05:18 2021 +0200  
  
version 1  
(END)
```