Assignment 2 report KPPDEA001 23 March 2022

Goal:

The goal of this experiment was to demonstrate the performance of AVL trees given different permutations of input data. The experiment displays the difference in number of operations taken to complete the search and insert algorithms using varied degrees of randomised data.

In order to execute the experiment, I have created a programme called AVLExperiment. The main class consists of four methods, a ReadFile method, a shuffle method, a load tree method and a HowRandom method.

The ReadFile method simply reads the data from the given CSV file into an array. The Shuffle method then n items in the array based on a given parameter. Once the array is shuffled to the specified degree, the data is loaded into an AVL tree using the LoadTree method and the find algorithm is performed on every item of data.

The operations are counted in the AVLTree class using integer variables which are incremented per operation. The results of the experiment are loaded into a separate array as a string. The string contains the number of operations taken to perform the search and find tasks as well as how random the data was that these tasks were performed on. This is given by the number of items that are out of order which is received from the HowRandom method.

Randomisation:

In order to randomise the data, I have created a method called shuffle. The method uses an algorithm which operates on the given number of items. The algorithm loops through the array n times and performs the following operations.

1. Generates a “random” number using the Math.random function which will be used as an index.
2. Stores the current index value in a temporary variable.
3. Copies the value at the index of the “random” number that was generated and stores it in the current index.
4. Stores the item of the temporary variable in the index of the “random” number.

In summary, these steps take the current index and swap its value with a value in a random index in order to shuffle the items in the array.

This algorithm is good due to its simplicity. The code is only several lines, and it is easily understandable by anyone. The algorithm is a very efficient way to make sure most, if not all items in the array ends up in a different position.

Results:

No randomisation (level 0)

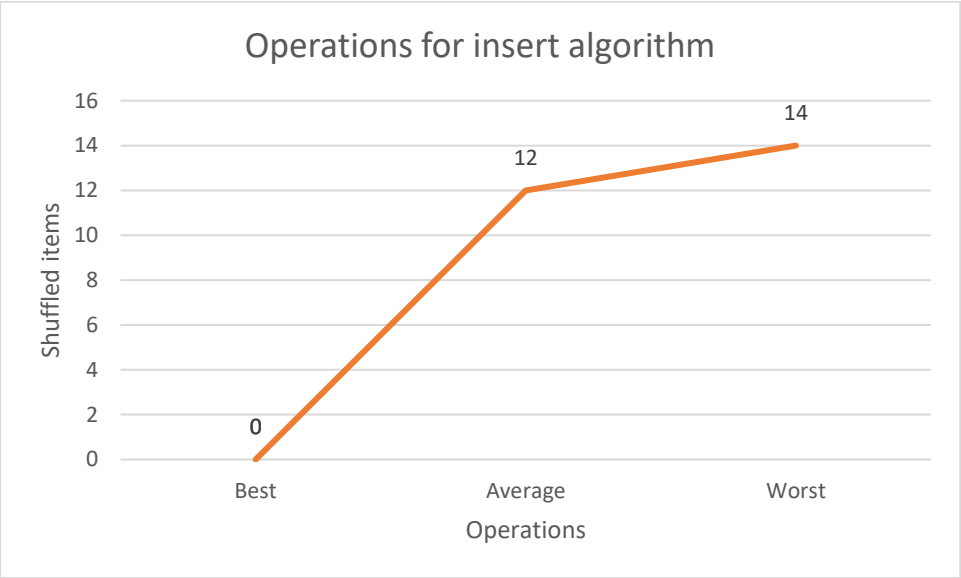
```
number of shuffled items: 0, Insert operations: 122657, Find operations  122613
Max insert operations: 14 Min insert operations: 0 Average insert operations: 12
Max find operations: 14 Min find operations: 1 Average find operations: 12
```

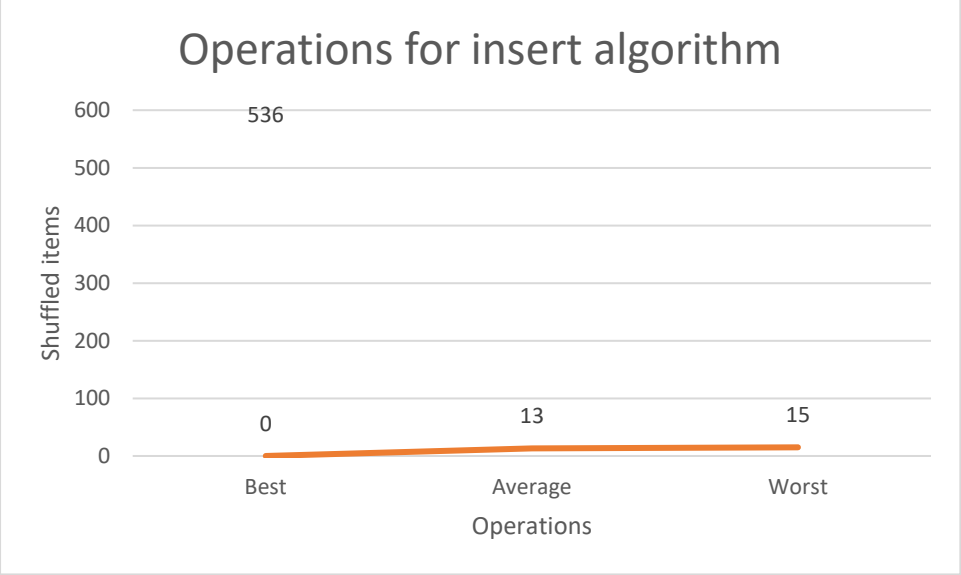
Level 10 of randomisation

```
number of shuffled items: 384, Insert operations: 127853, Find operations  123565
Max insert operations: 14 Min insert operations: 0 Average insert operations: 12
Max find operations: 14 Min find operations: 1 Average find operations: 12
```

Level 20 of randomisation

```
number of shuffled items: 536, Insert operations: 130937, Find operations  127515
Max insert operations: 15 Min insert operations: 0 Average insert operations: 13
Max find operations: 15 Min find operations: 1 Average find operations: 12
```





The results of the experiment demonstrate the efficiency of AVL trees. Due to the properties of an AVL tree of having the difference in height between sub trees less than or equal to one, the tree is constantly rebalanced and therefor will never become a linear data structure. Due to this fact it does not matter what order the items are inserted or searched for as the average and worst case time complexities will be O(log n), this is far more efficient then a regular binary search tree which has worst case time complexities of O(n). The graphs above show the small amount of change between number of operations for the insert algorithm with different permutations.

Creativity:

For this assignment I struggled to find a way in which to be creative however I have I feel as though I have performed the required tasks and I have used simple and easy to understand algorithms to do so and not over complicate the process. my experiment results or printed neatly to a text file for later review as well as to the terminal for instant viewing.

Git log:

```
7.      commit c4fb2224ed265f931d96af173f3b67b1728ab237
        Author: Dean <dkopping9@gmail.com>
        Date:   Thu Mar 17 12:43:56 2022 +0200

            counting find operations added however not fully operational

6.      commit fc1910966e125624cf9f46d505f1fd210e3df55b
        Author: Dean <dkopping9@gmail.com>
        Date:   Mon Mar 14 12:42:00 2022 +0200

            Experiment not prints number of insert operations for each degree of 20 randomisations

5.      commit a1bc8263e215698c6a6f68195821f7ac253e4b4b
        Author: Dean <dkopping9@gmail.com>
        Date:   Mon Mar 14 11:31:35 2022 +0200

            Updated classes

4.      commit d765a3045e0e4d00271b225a625dbd320243166d
        Author: Dean <dkopping9@gmail.com>
        Date:   Sun Mar 13 21:36:44 2022 +0200

            Empty shuffle function added for the array

3.      commit def1e4149ad2e500912cc0b66a00eedcc59c5d03
        Author: Dean <dkopping9@gmail.com>
        Date:   Thu Mar 10 22:14:28 2022 +0200

            Read file working

2.      commit e5de070a6e758533536befb5ce21fbf3ccb67ecb
        Author: Dean <dkopping9@gmail.com>
        Date:   Thu Mar 10 22:11:00 2022 +0200

            Read file method partially working and javadoc used

1.      commit e5d5ddd271391aaa5c602773922260c14463e179
        Author: Dean <dkopping9@gmail.com>
        Date:   Thu Mar 10 21:44:34 2022 +0200

           All files added to git repository and make file operational
```

```
12.     commit c7b740bbfe996ae5f129dfc713930901b0547c2e
        Author: Dean <dkopping9@gmail.com>
        Date:   Tue Mar 22 14:41:05 2022 +0200

            aveage case values added

11.     commit f56e60eae253af0f7ca1abcceb40e569383718d9
        Author: Dean <dkopping9@gmail.com>
        Date:   Tue Mar 22 14:31:41 2022 +0200

            added functionality to find min and max search values

10.     commit c8d28e430332d92b75ebb07ba3f62556d36fc551
        Author: Dean <dkopping9@gmail.com>
        Date:   Tue Mar 22 11:58:50 2022 +0200

            Programme now successfully prints result to "Experiment results" file

9.      commit d85a099dc034b8e17f721afc32a73ea1bf92ff90
        Author: Dean <dkopping9@gmail.com>
        Date:   Tue Mar 22 11:41:03 2022 +0200

            updated shuffling, more effective

8.      commit 3b64ed8047cff28d715b296e435f8384312e61fc
        Author: Dean <dkopping9@gmail.com>
        Date:   Tue Mar 22 11:31:52 2022 +0200

            Everything working properly aside from operation counter
```