Data Structures

Themba Shongwe (SHNTHE021)

Data Structures

In today's world, many systems make use of Artificial intelligence. Artificial intelligence has a become a very huge resource to many companies due to the vast information it holds. This information stems from developers feeding A.I systems data and information collected over decades all around the world, to make one program that makes it easier for users to access information from anywhere in the world. As a Computer Scientist, it imperative that I understand how the data that is collected, is stored and retrieved by users. I also understand the need to provide factual information and also provide an option to update misinformation/outdated information. In this section, I was exploring two commonly used data structures and how well they handle real-world data supplied through text files.

AIM:

The aim for this is assignment is to develop a Java program which allows manipulation (i.e. querying and updating) to a dataset provided by a user to the program. The program must be able to manage loading a text File containing a knowledge base derived from GenericsKB, a dataset of general knowledge statements. The program must be able to give users information that is deemed correct by the knowledge. The other aim for the assignment is to evaluate prior knowledge of arrays and binary search tree, to get the practically evaluate which is more effective when it comes to handling, searching, updating, and adding elements of the data.

METHOD:

Apparatus used:

- Computer (supporting wsl)
- Files with data to use for test cases.

My thought process for the program was to make a program which will take in each line and separate each data entry by line. This served convenient as I know every line has the same data fields (term, statement, and confidence score) so it then made it easier to think up ways to access the data that I would need to handle the different userrequests which would be to search, update and add new data to the knowledge.

OOP DESIGN(overview):

Classes created:

- ***mainInterface.java:***
  This is the main class which acts as the interface connecting the user to the knowledge base. It is the umbrella class which connects two major classes, GenericsKbArrayApp.java and GenericsKbBSTApp.java, which process the user interaction differently depending on which data structure the user wanted to use, an array or binary search tree respectfully. The mainInterface prompts the user to choose which data structure they like to use, then will generate questions whilst knowing which class is being used.
- ***GenericsKbArrayApp.java:***
  This class is used to process the knowledge into an array data structure. I first loaded the knowledge base into the array by making each line into a GenericsKbArrayApp object which has a term, statement and confidence score.I then included getter methods to access the data easily. This then made the searching portion of the program easier as I would just return the required statement or confidence score if the term was found. For the Array App, I could not add an element, so in this program I provided an update feature for specified term(only updates if confidence score is higher).
- ***GenericsKbBSTApp.java:***
  This part of the program uses the Binary Search Tree as its data structure. It incorporate the binary Search tree, binary tree, binary treenode , binary queue and binary queuenode classes provided by the university (University Of Cape Town).It also utilises the datastore class which will explain in the next bullet point.The tree contains nodes which all contain data (a single line in the knowledge base) and the reference to the next tree node. This make it easier to traverse and search for required term statements and confidence score.
- ***datastore.java:***
  The datastore class is used for the creation of binary tree nodes(for the most part). The class creates objects depending on the amount of data provided, that contain a term, statement and confidence score with their respective getter methods.This then makes it easier to store a single line of data as it is easier to access an object.

TESTING:

ArrayApp:

```
Welcome to your General Knowledge Box :)
----------------------------------------
Choosebetween using an Array and using a Binary Search Tree.
1.Use Array.
2.Use Binary Search Tree.
Choice: 1
You chose an Array. :)
Choose an action from the menu:
```

prompting user to choose data structure.

```
Welcome to your General Knowledge Box :)
----------------------------------------
Choosebetween using an Array and using a Binary Search Tree.
1.Use Array.
2.Use Binary Search Tree.
Choice: 1
You chose an Array. :)
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 1
Enter file name: GenericsKB.txt
Knowledge base loaded successfully.
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 3
Enter the term to search: maple syrup
Statement Found: Maple syrup is syrup.(Confidence score: 1.0)
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 5
Well hope you enjoyed your time here, Hope to see you again soon :)
```

Uploading knowledge base and searching in the array part of the program.

```
1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 2
Enter the term: root hair
Enter statement: something to do with hair
Enter confidence score: 1.0
Statement for root hair has been updated.
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 3
Enter the term to search: root hair
Statement Found: something to do with hair(Confidence score: 1.0)
Choose an action from the menu:
```

updating statement in the array.(also applies for binary search tree)

Binary Search Tree:

```
Welcome to your General Knowledge Box :)
----------------------------------------
Choosebetween using an Array and using a Binary Search Tree.
1.Use Array.
2.Use Binary Search Tree.
Choice: 2
You chose an The Binary Search Tree. :)
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 1
Enter file name: GenericsKB.txt
Knowledgebase loaded successfully.
Choose an action from the menu:
```

prompting user to choose data structure followed by uploading knowledge base.

```
1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 3
Enter the term to search: criminologist
Statement Found: Criminologists are workers.(Confidence score: 1.0)
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 4
Enter the term to search: criminologist
Enter statement to search for: criminologists are the best!!
Match not found. :(
Choose an action from the menu:
```

Searching

```
1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 2
Enter the term: fail
Enter statement: to not meet pass requirements.
Enter confidence score: 1.0
Statement for fail has been added.
Choose an action from the menu:

1.Load a knowledge base from a file
2.Add a new statement to the knowledge base
3.Search for a statement in the knowledge base by term
4.Search for a statement in the knowledge base by term and sentence
5.Quit
Enter your choice: 3
Enter the term to search: fail
Statement Found: to not meet pass requirements.(Confidence score: 1.0)
Choose an action from the menu:
```

adding to the tree and search if it really added.

CREATIVITY:

- For the creativity portion, I did not include much, all that I feel constitutes creativity would be me adding emoticons at the end of response statements. This is because I wanted to give a sense of relaxation to users to feel welcomed. Also including the "**welcome to the General knowledge box**" statement and prompting the user to choose preferred structure.

- Another thing that I believe shows creativity would be me doing a text terminal mainInterface to accommodate for me not having the time to actually make a GUI.

Git Commit History(From latest to oldest):

1. **Commit:** 64975b42c74d5ba466d6cfa0f19739a924f2d3bd
   Author: Themba Shongwe
   Date: Mon Mar 17 14:06:29 2025 +0000
   Message: Addition of Makefile.

2. **Commit:** 18d6b26978646917ebf0078d372415be4c53d42b
   Author: Themba Shongwe
   Date: Mon Mar 17 08:58:51 2025 +0000
   Message: Finished programs with mainInterface which will act as the interface between the user and the data structure programs.

3. **Commit:** dc67e1dfc1e023fc4dbc2bcf8d28f51ca9d65638
   Author: Themba Shongwe
   Date: Sun Mar 16 22:35:26 2025 +0000
   Message: GenericsKbArrayApp and dataStore Java files commented and with their javadocs.

4. **Commit:** b97bad615edb0bdedd359f3b08e89e27f4b72861
   Author: Themba Shongwe
   Date: Sun Mar 16 22:10:17 2025 +0000
   Message: Commented GenericsKbBSTApp along with its Javadoc.

5. **Commit:** 480f8ef47111edf303059b08d859ae7bf8b28052
   Author: Themba Shongwe
   Date: Sun Mar 16 20:07:05 2025 +0000
   Message: GenericsKb apps, fully working. Have to add javadocs and comments.

6. **Commit:** 19dc719394bc245e129ab861a6be9c5ad14b7cdf
   Author: Themba Shongwe
   Date: Sun Mar 16 17:29:17 2025 +0000
   Message: Updated and commented GenericsKbArrayApp, along with unfinished GenericsKbBSTApp, with its project files.

7. **Commit:** 744f2df0ee3fbc970300783eaf01e4f599357d07
   Author: Themba Shongwe
   Date: Sat Mar 15 18:54:34 2025 +0000
   Message: First commit, GenericsKBArray.java is uncommented.

8. **Commit:** 7467a329f8fbd1237eeabf89472f6b5568f92982
   Author: Themba Shongwe
   Date: Fri Mar 14 21:46:01 2025 +0200
   Message: First time adding to git, kinda nervous (INITIAL COMMIT).

CONCLUSION:

Through this implementation, I was able to evaluate the strengths and weaknesses of both data structures. The array-based approach provided a simple yet inefficient way to store and update data, as searching and updating required linear traversal. On the other hand, the BST-based approach demonstrated better efficiency in searching and updating, thanks to its structured hierarchy. However, because the BST was unbalanced, performance could degrade in the worst case.
This project reinforced my understanding of data structures, file handling, and object-oriented programming in Java. Additionally, implementing a structured user interface improved the usability of the application. Future improvements could include balancing the BST, optimizing file I/O operations, and integrating more advanced search functionalities.

Overall, this assignment provided valuable hands-on experience in designing, implementing, and evaluating data structures in a real-world context.