2.0 Individual Report – CHAN SEOW FEN (0207368)

Data structure refers to a specific format for data organization, processing, retrieval, and storage. Data structure allows users to quickly access and process the data they require. To illustrate it, there are two types of data structure, which is primitive data structure and non-primitive data structure (Loshin and Lewis, 2021). In short, primitive data structures is directly controlled by machine instructions while non-primitive data structures allow storing variable in multiple data type. The examples of primitive data structures are int, string, char, float and double. Non-primitive data type can be further divided into linear data structure and non-linear data structure. A linear data structure is made up of data elements that are organized in sequence, with each element connected to the elements before and after it while a non-linear data structure has no fixed order in which its components are connected, and each element can have multiple paths to other elements (Vishnu R, 2021). The examples of linear data structure are Array, ArrayList, Linked list, queue and stack while the examples of non-linear data structure are tree and graph. The data structure that has been chosen for developing the program is ArrayList. The ArrayList data structure is a resizable or dynamic array data structure that stores elements in sequential order and can be increased or reduced in size by adding or removing elements (Adservio.fr, 2023). One of the advantages of ArrayList are it does not require to mention the size when declaring the ArrayList. In addition, it benefits in the way that we can insert different types of variables into the ArrayList. Furthermore, elements can be added or removed from a particular position. Moreover, it can handle multiple elements that are null (pramodbablad, 2014). Additionally, using the ArrayList.get(element) method to get a particular element in it is extremely fast. On the other side, the disadvantage of ArrayList are slow insertion or deletion of data as updating the list need to shift the data. Furthermore, memory wastage occurred because larger components of a list require substantial contiguous blocks of memory. Moreover, resizing an ArrayList when it hits its initial capacity of 10 is a more expensive procedure because the elements are copied from the old to the new space with 50% more capacity (DevGlan, 2019).

ArrayList can be implemented by using ArrayList<String> syntax. Elements is inserted using add() method, accessed by using get() method and deleted by using remove() method. The application of ArrayList is used when need to store and manipulate large amounts of data especially when the size is not known beforehand. It is also been used when intend to insert duplicate elements into the list. Moreover, it is used when null elements needed to be inserted (Easy, 2020). Furthermore, it is also used to implement other data structure such as stack, queue and hash table. The memory used in ArrayList is totally depends on the size of the ArrayList as well as the data type stored in the ArrayList hence its space complexity is O(n), where the n is the number of elements in the list. In addition, the size of the ArrayList can be obtained using size() method while the capacity can be obtained using capacity() method. On the other hand, the time complexity of ArrayList varies depending on the specific operation being done. The time complexity is shown in the Table 1.2. For justification, operation 1 has an O(1) time complexity because ArrayList stores the elements in an array, and accessing an element by index is simply a matter of indexing into the array. Because all the elements after the specified index must be shifted to make space for the new element, operation 2 has an O(n) time complexity. The time complexity of operation 3 is O(1) on average, but O(n) in the worst situation. When the ArrayList is complete, it creates a new array that is twice the size of the previous array and copies its elements to the new array. As a result, this operation is an amortized O(1) process. Due to the reason that all the elements after the given index must be shifted to fill the gap left by the removed element, operation 4 has an O(n) time complexity. As it simply entails removing the last element from the array, operation 5 has a time complexity of O(1). Finally, because ArrayList does not have a built-in mechanism for looking for an element, the time complexity of Operation 6 is O(1).

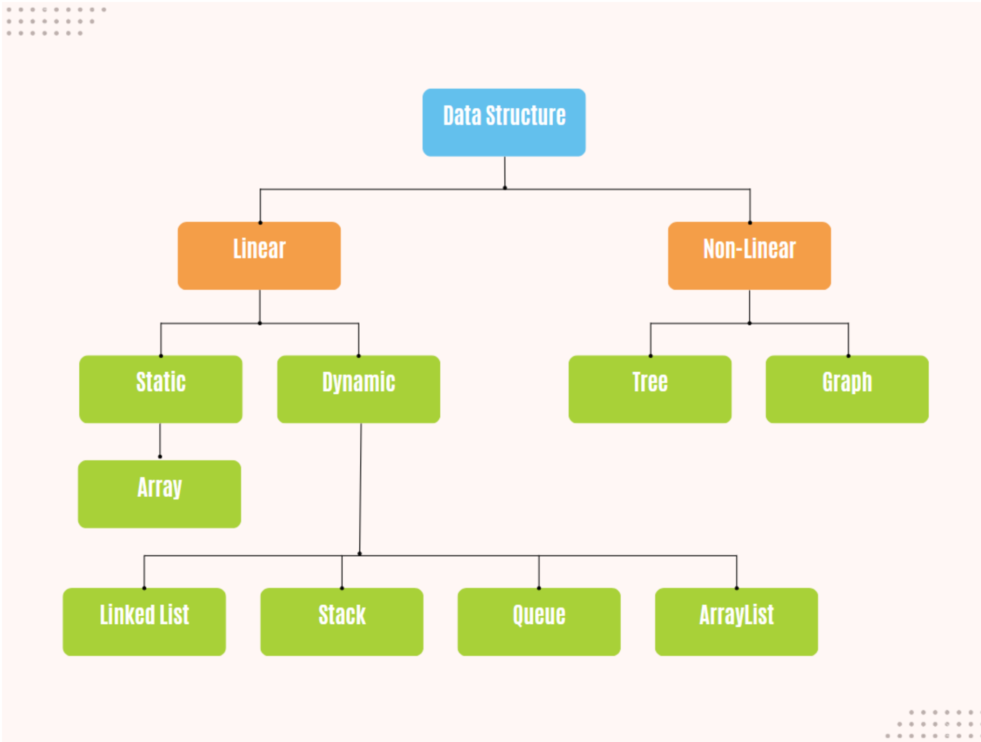

Figure 1.1 Type of Data Structure

	Operation	**Time Complexity in terms of Big-O notation**
1.	Accessing an element by index	O(1)
2.	Inserting an element at a specific index	O(n)
3.	Inserting an element at the end of the ArrayList	O(1) (amortized time)
4.	Removing an element from a specific index	O(n)
5.	Removing an element from the end of the ArrayList	O(1)
6.	Searching for an element	O(1)

Table 1.2 Time Complexity of ArrayList

3.0 Reference List

Adservio.fr. (2023). *ArrayList vs. LinkedList in Java: What I Need to Know*. [online] Available at: https://www.adservio.fr/post/arraylist-vs-linkedlist-in-java-what-i-need-to-know [Accessed 21 Mar. 2023].

DevGlan (2019). *Pros and Cons of Collection Types in Java | DevGlan*. [online] devglan. Available at: https://www.devglan.com/corejava/pros-and-cons-collection-java [Accessed 21 Mar. 2023].

pramodbablad (2014). *Advantages Of Using ArrayList Over Arrays*. [online] Java Concept Of The Day. Available at: https://javaconceptoftheday.com/advantages-of-using-arraylist-over-arrays/ [Accessed 21 Mar. 2023].

Easy, S. (2020). *Realtime Use of ArrayList in Java with Example*. [online] Scientech Easy. Available at: https://www.scientecheasy.com/2020/09/use-of-arraylist-in-java.html/ [Accessed 21 Mar. 2023].

Loshin, D. and Lewis, S. (2021). *data structures*. [online] Data Management. Available at: https://www.techtarget.com/searchdatamanagement/definition/data-structure [Accessed 21 Mar. 2023].

Vishnu R (2021). *What is Data Structure: Need, Types & Classification*. [online] Great Learning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/data-structure-tutorial-for-beginners/ [Accessed 21 Mar. 2023].