Advanced Programming Assignment 2

**Question 1:**

|  |  |  |
| --- | --- | --- |
| Differences | ArrayList | Vector |
| Synchronized | Not synchronized. Collections.synchronizedList is normally used at the time of creation of the list to avoid any accidental unsynchronized access to the list. | Synchronized by default. |
| Data growth | **ArrayList increases its array size by 50 percent.** | **A Vector defaults to doubling the size of its array** |
| Legacy class | not a legacy class, it is introduced in JDK 1.2 | is a legacy class. |
| Element traversal | Uses Iterator interface to traverse the elements. | Uses Enumeration interface to traverse the elements. But it can use Iterator also. |

**Question 2:**

|  |  |  |
| --- | --- | --- |
| Differences | Hashset | Sortedset |
| Ordering | Unsorted | Sorted |
| Contiguous storage | Yes | No |
| Lookup Efficiency | O(1) | O(logn) |
| **Data Structure** | Uses a Hash-table | Uses a balanced binary tree |

**Question 3:**

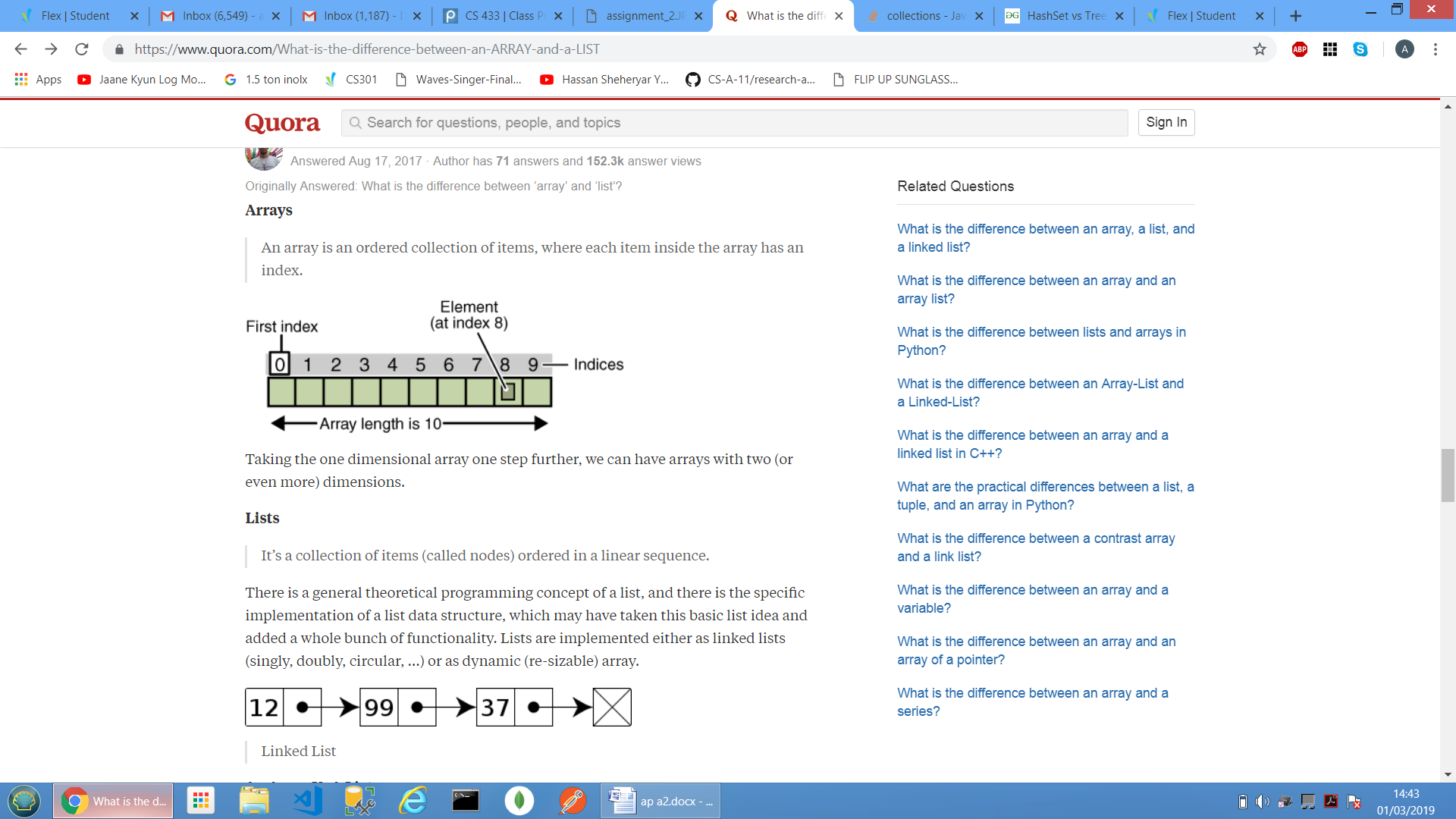
|  |  |  |
| --- | --- | --- |
| Differences | Hashset | TreeSet |
| Complexity | O(1) complexity to add or remove element | O(logn) add or remove complexity |
| Null objects | Allows null objects | Does not allow null objects, throws nullpointerException |
| Ordering | No order guarenteed | Maintains sorted order |

**Question 4:**

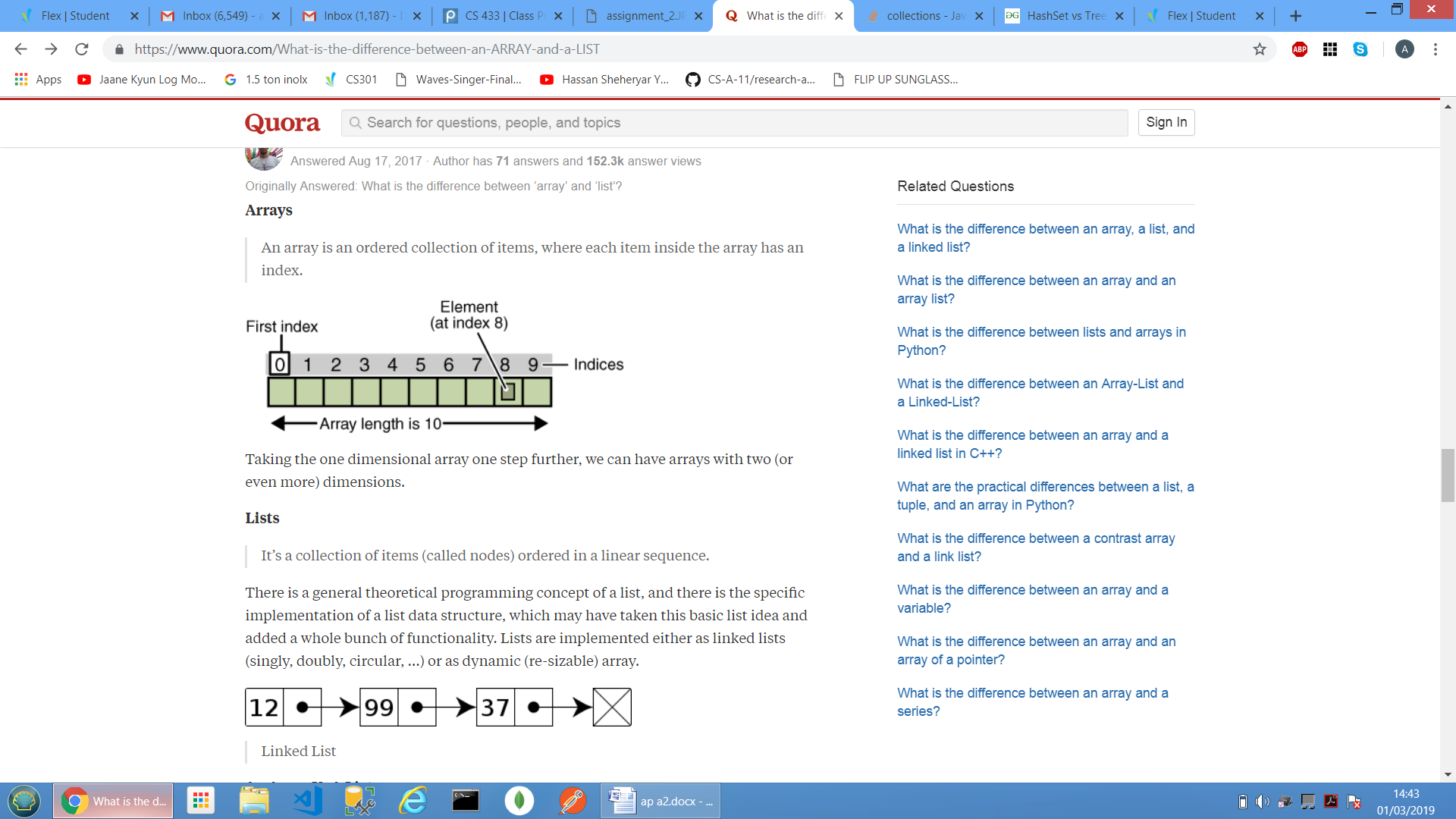
List is an interface whereas ArrayList is an implementation of List. So comparing Array and LinkedList.

|  |  |  |
| --- | --- | --- |
| Differences | Array | List (LinkedList) |
| Memory | Requires contiguous memory | Does not require contiguous memory |
| Accessing elements | Allows Random access and sequential access | Allows only Sequential access |
| Numeric indexing | Supports numeric indexing | Does not support numeric indexing |
| Element type | Collection of homogenous elements | Allows heterogeneous data as well. |
| Allocation | Static allocation | Dynamic allocation |

Array:



List:



**Question 5:**

|  |  |  |
| --- | --- | --- |
| Differences | List | Set |
| Order | Ordered | Depends upon implementation |
| Duplicates | Allows duplicates | Doesn't allow duplicates |
| Null values | Permits number of null values | Permits only one null value. |
| New methods | Defined inside List interface | Not defined inside Set interface use Collection interface methods only with Set subclasses. |

**Question 6:**

NavigableMap implements interface map which is a data structure that associates its elements with certain keys so that these elements could be obtained by that keys. whereas NavigableSet implements interface set where both do not allow duplicates (i.e. keys in maps).

Both of them maintain a sorted order as they inherit from SortedMap and SortedSet respectively and provide methods for navigating over elements.