Collection framework

Additional depth knowledge :

1. Internal implementation of each collection
   1. Arryalist - array
   2. Linkedlist - doubly linked list
   3. Hashset - Hashmap
   4. Treeset - treemap
   5. Hashmap - hashtable
   6. Treemap - red-black tree - self balancing binary tree
   7. Priority queue - binary heap
   8. Hashtable - an array of buckets, where each bucket contains a list of entries.
2. Null values allowed
   1. Arraylist - multiple
   2. Linkedlist - multiple
   3. Hashset - only one
   4. Treeset - not allowed
   5. Priority queue - not allowed
   6. Hashmap - only one null key but multiple null values
   7. Treemap - not allowed
   8. Hashtable - no null key or value
3. Fail-fast ietrators and fail-safe iterators   
   Fail fast ietrators thorw ConcurrentModificationException if underlying collections modifies after iterator is created.  
   Fail safe iterators does not throw any exception even if if underlying collection gets modified e.g. ConcurrentHashMap
4. forEach method is available to iterate the collection  
   nums.forEach(t -> System.out.println(t));  
   this method needs implementation of Consumer interface having method as void accept(<type>) where type as that of type of collection
5. The Arrays class in java.util package is a part of the Java Collection Framework. This class provides static methods to dynamically create and access Java arrays. It consists of only static methods and the methods of Object class  
   asList()  
   sort()  
   binarySearch()  
   fill()  
   compare()  
   System.out.println(Arrays.toString(new int []{4,7,2,9,3}));
6. Arrays.sort() is a method residing in Arrays class. It is used to sort the Array passed to it. It can be integer array, float array, String array, Array of objects etc. The time complexity for this method is O(n log n) as it runs TimSort in background. TimSort algorithm makes use of the [Insertion sort](https://atechdaily.com/posts/Algorithm-for-Insertion-Sort) and the [MergeSort](https://atechdaily.com/posts/Merge-Sort-Algorithm-or-Pseudocode-in-Java) algorithms.  
   Collections.sort() is used to sort an object which extends List interface. ArrayList and LinkedList extend List interface, so we can sort them using Collections.sort. Collections.sort() has a time complexity of O(n log n) as it run merge sort in background.