1. **public static <T extends Comparable<? super T>> void sort(List<T> list)**

   Public - is the access modifier
   static - means the method is at class level and you don't need to create an instance of this method to execute it.
   <T extends Comparable<? Super T>> - this method is of generic type T which implements the comparable interface for the compareTo() method. <? super T> is a wildcard that allows T or any superclass of T to be used as the type argument for Comparable. This is a way to ensure that you can compare elements of type T and types that are superclasses of T.
   Void - is the return type of this function which doesn't return anything
   Sort - is the name of the function
   (List<T> list)- is the parameters that sort takes in; where List<T> is the type of parameter that it accepts and list is the name of the parameter.

2. **public static <T> void sort(List<T> list, Comparator<? super T> c)**

   Public - is the access modifier
   static - means the method is at class level and you don't need to create an instance of this method to execute it.
   <T>- this method is also generic of the type T.
   Sort- is the name of the method.
   (List<T> list, Comparator<? super T> c)- is the parameters that the function sort takes in which are:
   - **list -** which is of the type List<T> ie. list of generic type
   - **c -** which is of the type Comparator<? super T> ie. of the type comparator and is a a generic type and a wildcard which means it can be anytype of type T and that extends the super class of T.

   Void - is the return type of this function which doesn't return anything

3. **public static <T> int binarySearch(List<? extends Comparable<? super T>> list, T key)**

   Public - is the access modifier
   static - means the method is at class level and you don't need to create an instance of this method to execute it.
   <T>- means that this method is of generic type T.
   int - is the return type of this method ie. an integer.
   binarySearch - is the name of the method.
   (List<? extends Comparable<? super T>> list, T key) - are the parameters that binarySearch takes in which are:
   - **list - is of the type List which accepts any generic type T and the implements comparable to compare the values inside of it. It compares the values inside the list which are of type T as well as the superclass of T.**

- **key - the key parameter is of the type T which is the generic type defined.**

4. **public static void shuffle(List<?> list)**

   Public - is the access modifier
   static - means the method is at class level and you don't need to create an instance of this method to execute it.
   Void - is the return type of this function which doesn't return anything
   Shuffle - is the name of the method.
   (List<?> list)- is the parameter that the method takes in which is list and the type of it is a List<?> which means it can accept elements for the list of any type which is not parameterized in any way.

5. **public static <T> void copy(List<? super T> dest, List<? extends T> src)**

   Public - is the access modifier
   static - means the method is at class level and you don't need to create an instance of this method to execute it.
   <T>- means that this method is of generic type T.
   Void - is the return type of this function which doesn't return anything
   Copy- is the name of the method
   (List<? super T> dest, List<? extends T> src)- these are the two types of the parameters that the method takes in which are:
   - **dest-** is of the type List<? super T> which takes in the elements of all types of the type T and its super class.
   - **src -** is another List<? extends T> which means it is of type list that accepts any type T or any of its sub type.