CSCI3055 Assignment 1 Part 3

(3.1) We can use the filter function to divide the input list to two sublist based on a pivot point. All the number less than the pivot will be on left while all the number greater than then pivot will be on the right.

(3.2) Java doesn't fully support curried function. However, we can do it by compare every element in the list to a pivot point and add it the left or right. Then, we can do the quicksort recursively. Finally, we can combine all the left, middle and right list together.

```
Public <K extends Comparable> List<K> quicksort(List<K> input) {
        if (input.isEmpty()){
                return input
        } else {
                K pivot = input.get(0);
                List<K> left = new LinkedList<K>();
                List<K> right = new LinkedList<K>();
                List<K> middle = new LinkedList<K>();
                List<K> sorted = new LinkedList<K>();
                for (K k : input) {
                        if (k.compareTo(pivot) < 0) {
                                left.add(k);
                        } else if (k.compareTo(pivot) > 0) {
                                right.add(k);
                        } else {
                                middle.add(k);
                        }
                }
                left = quicksort(left);
                right = quicksort(right);

                final.addAll(left)
                final.addAll(middle);
                final.addAll(right);

                return final;
        }
}
```