Method under Test

```
public static <E extends Comparable<? super E>>  List<E>
almostSortedListSort(List<E> list, int k) {
   //preconditions checking
   Sorting.verifyListNonNull(list);
   Sorting.verifyDistortionFactor(k);

   //let pq be a new priority queue
   PriorityQueue<E> queue = new PriorityQueue<>();

   //insert first k+1 elements into the queue
   if(!list.isEmpty()) { //1
       queue.addAll(list.subList(0,k+1)); //2
   }

   //let sortedList be a new list
   List<E> output = new LinkedList<>();

   //insert remaining elements from list into the queue
   //remove head of priority queue and add to the output list
   for(int i = k+1; i < list.size(); i++) { //3
       queue.add(list.get(i)); //4
       output.add(queue.remove()); //5
   }

   //remove remaining elements of priority queue and add to output list
   while(!queue.isEmpty()) { //6
       output.add(queue.remove()); //7
   }
   //return output list as sorted list
   return output;
}
```

Test Conditions

Goal	Notes	Condition
(CC1) Code Coverage	1, if, 2	k >= 0

(CC2) Code Coverage	3, if, 4+5	k+1 < list.size()
(B1) Branch Coverage	3 false	k + 1 >= list.size()
(CC3) Code Coverage	6, if, 7	!queue.isEmpty()
(B2) Branch Coverage	6 false	queue.isEmpty()
(b1) Boundary	3	k+1 = list.size()
(b2) Boundary	3	k+1 < list.size()
(b3) Boundary	3	k+1 > list.size()
(B3) Branch Coverage	1 false	list.isEmpty()

Unless stated otherwise, it is assumed that list ≠ Ø and 0 <= k <= 100

Tests

Test Condition	Conditions Satisfied	Assertions
(1) list = {x1, x2, x3, …}, k >= 1	CC1, CC2, CC3, b2	list = {x1, x2, x3, …} sorted
(2) list = {x1, x2}, k = 1	CC1, B1, CC3, b1	list = {x1, x2} sorted
(3) list = {x1, x2, …}, k = 0	CC1, B1, CC3, b1	list = {x1, x2, …} sorted
(4) list = {}, k = 0	CC1, B1, B2, B3, b3	list = {}

Good Data Tests:

- Nominal size of list with proper distortion factor
- Minimum size of list → list is empty
- Maximum size of list → list with k=100

Bad Data Tests:

- Lists with improper corresponding distortion factors
- Lists that are null
- Lists with null values
- Lists with negative k values or where k > 100

Stress Test

Create a list with a distortion factor of 100 and with a large number of elements, for example one thousand elements. Sort the list using the algorithm and compare the resulting output with a sorted version of the input list. Repeat 50 times.