Array Selector

Problem Overview

This assignment focuses on implementing the methods of a class much like `java.util.Arrays`. The `Selector.java` file defines a class with static methods designed to provide useful functionality on arrays. Each method of `Selector` is very clearly specified, is independent of the other methods in the class, and is designed to provide relatively simple functionality. So, this is a great context for practicing what we are discussing in lecture – systematic, disciplined development and test-based verification.

The `Selector` class

You must correctly implement all the method bodies of the provided `Selector` class. Your implementation must adhere *exactly* to the API of the `Selector` class, as described in the provided source code comments and as described below.

```
public final class Selector {
   public static int   min(int[] a)
   public static int   max(int[] a)
   public static int   kmin(int[] a, int k)
   public static int   kmax(int[] a, int k)
   public static int[] range(int[] a, int low, int high)
   public static int   ceiling(int[] a, int key)
   public static int   floor(int[] a, int key)
}
```

The `min` method.

This method selects the minimum value from a given array. If the array is null or has zero length, this method throws an IllegalArgumentException. The array is not changed by this method.

Examples:

a[ ]	min(a)
[2, 8, 7, 3, 4]	2
[5, 9, 1, 7, 3]	1
[8, 7, 6, 5, 4]	4
[2, 8, 8, 7, 3, 3, 4]	2

The `max` method.

This method selects the maximum value from a given array. If the array is null or has zero length, this method throws an IllegalArgumentException. The array is not changed by this method.

Examples:

a[ ]	max(a)
[2, 8, 7, 3, 4]	8
[5, 9, 1, 7, 3]	9
[8, 7, 6, 5, 4]	8
[2, 8, 8, 7, 3, 3, 4]	8

The `kmin` method.

This method selects the k^{th} minimum value from a given array. A value is the k^{th} minimum if there are exactly $k - 1$ values less than it in the array. If the array is null, has zero length, or if there is no k^{th} minimum value, this method throws an IllegalArgumentException. Note that there is no k^{th} minimum value if k is less than 1, k is greater than the number of elements in the array, or if k is greater than the number of distinct values in the array. The array is not changed by this method.

Examples:

a[ ]	k	kmin(a, k)
[2, 8, 7, 3, 4]	1	2
[5, 9, 1, 7, 3]	3	5
[8, 7, 6, 5, 4]	5	8
[2, 8, 8, 7, 3, 3, 4]	3	4

The `kmax` method.

This method selects the k^{th} maximum value from a given array. A value is the k^{th} maximum if there are exactly $k - 1$ values greater than it in the array. If the array is null, has zero length, or if there is no k^{th} maximum value, this method throws an IllegalArgumentException. Note that there is no k^{th} maximum value if k is less than 1, k is greater than the number of elements in the array, or if k is greater than the number of distinct values in the array. The array is not changed by this method.

Examples:

a[ ]	k	kmax(a, k)
[2, 8, 7, 3, 4]	1	8
[5, 9, 1, 7, 3]	3	5
[8, 7, 6, 5, 4]	5	4
[2, 8, 8, 7, 3, 3, 4]	3	4

The `range` method.

This method returns an array of all values i from a given array such that $low \leq i \leq high$, including duplicate values. (Note that *low* and *high* do not have to be actual values in the given array, and the order of the resulting values is irrelevant.) The length of the returned array is the same as the number of values i that meet the criterion. If there are no values i that meet the criterion, this method returns a zero-length array. If the given array is null or has zero length, this method throws an IllegalArgumentException. The given array is not changed by this method.

Examples:

a[ ]	low	high	range(a, low, high)
[2, 8, 7, 3, 4]	1	5	[2, 3, 4]
[5, 9, 1, 7, 3]	3	5	[3, 5]
[8, 7, 6, 5, 4]	4	8	[8, 7, 6, 5, 4]
[2, 8, 8, 7, 3, 3, 4]	3	7	[7, 3, 3, 4]

The `ceiling` method.

This method returns the smallest value i in a given array such that $i \geq key$. (Note that *key* does not have to be an actual value in the given array.) If the given array is null or has zero length, or if there is no qualifying value i, this method returns an IllegalArgumentException. The given array is not changed by this method.

Examples:

a[ ]	key	ceiling(a, key)
[2, 8, 7, 3, 4]	1	2
[5, 9, 1, 7, 3]	7	7
[8, 7, 6, 5, 4]	0	4
[2, 8, 8, 7, 3, 3, 4]	5	7

The `floor` method.

This method returns the largest value i in a given array such that $i \leq key$. (Note that *key* does not have to be an actual value in the given array.) If the given array is null or has zero length, or if there is no qualifying value i, this method returns an IllegalArgumentException. The given array is not changed by this method.

Examples:

a[ ]	key	floor(a, key)
[2, 8, 7, 3, 4]	6	4
[5, 9, 1, 7, 3]	1	1
[8, 7, 6, 5, 4]	9	8
[2, 8, 8, 7, 3, 3, 4]	5	4

Notes and other requirements

Here are more specific requirements, notes, and suggestions.

- Read this handout carefully. Read the provided source code carefully. Ask questions on Piazza. Start early and be proactive.
- The constructor has been completed for you and must not be changed in any way.
- You may add any number of private methods that you like, but you may not add any public method or constructor, nor may you change the signature of any public method or constructor.
- You must not add any fields, either public or private, to the `Selector` class.
- You are allowed to use sorting **only** as part of your solution to `kmin` and `kmax`. You are not required to use sorting, but you are allowed to do so for these two methods **only**. If you use sorting, I strongly recommend the use of the `Arrays.sort` method.
- You must not import anything other than `java.util.Arrays`. If you do not use `java.util.Arrays`, then delete its import statement.