```java
package edu.neu.coe.info6205.sort.par;

import java.util.Arrays;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ForkJoinPool;

/**
 * This code has been fleshed out by Ziyao Qiao. Thanks very much.
 * CONSIDER tidy it up a bit.
 */
class ParSort {
    public static ForkJoinPool mPool;
    public static int cutoff = 1000;

    public static void sort(int[] array, int from, int to) {
        if (to - from < cutoff) Arrays.sort(array, from, to);
        else {
            // FIXME next few lines should be removed from public repo.
            CompletableFuture<int[]> parsort1 = parsort(array, from, from + (to - from) / 2); // TO IMPLEMENT
            CompletableFuture<int[]> parsort2 = parsort(array, from + (to - from) / 2, to); // TO IMPLEMENT
            CompletableFuture<int[]> parsort = parsort1.thenCombine(parsort2, (xs1, xs2) -> {
                int[] result = new int[xs1.length + xs2.length];
                // TO IMPLEMENT
                int i = 0;
                int j = 0;
                for (int k = 0; k < result.length; k++) {
                    if (i >= xs1.length) {
                        result[k] = xs2[j++];
                    } else if (j >= xs2.length) {
                        result[k] = xs1[i++];
```

```
31                      } else if (xs2[j] < xs1[i]) {
32                          result[k] = xs2[j++];
33                      } else {
34                          result[k] = xs1[i++];
35                      }
36                  }
37                  return result;
38              });
39
40              parsort.whenComplete((result, throwable
    ) -> System.arraycopy(result, 0, array, from,
    result.length));
41  //              System.out.println("# threads: "+
    ForkJoinPool.commonPool().getRunningThreadCount());
42              parsort.join();
43          }
44      }
45
46      private static CompletableFuture<int[]> parsort
    (int[] array, int from, int to) {
47          return CompletableFuture.supplyAsync(
48                  () -> {
49                      int[] result = new int[to -
    from];
50                      // TO IMPLEMENT
51                      System.arraycopy(array, from,
    result, 0, result.length);
52                      sort(result, 0, to - from);
53                      return result;
54                  }, mPool
55          );
56      }
57
58
59  }
```