Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University

Lab 4 – Merge Sort

Aim

Understand merge sort.

Resources

All Java files you need are found on ICE.

Merge sort Implementation

At this Lab session, we are going to implement merge sort and quick sort in java. You can refer to lecture notes 18.

Implement merge sort

Merge sort is a divide and conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. **The *merge ()* function** is used for merging two halves, it is the key process that assumes that two half of the arrays are sorted and merges the two sorted sub-arrays into one.

Download ***MergeSorter.java*** from ICE.

The static *sort (…)* method is the method to start sorting data.

The static *mergeSort (…)* is already completed. Have a look and try to understand it, refer to lecture 18.

Task: Please complete *merge (…)* method.

Note: you will need to use a temp array to keep the values. Once you finish merging all the elements into your temp array, copy your value back to the original data array.

Hint: use *System.arraycopy ()* is a bit faster than a for loop.

Once you finish your implementation, run file to test your sorting method. You can create more object arrays to test it.

Extra:

We can cut the running time of merge sort substantially with some carefully considered modifications to the implementation.

For example, we can use insertion sort for small subarrays. We can improve most recursive algorithms by handling small cases differently.

Please complete *insertionSort ()* method.

Update your *mergeSort (…)* method. When the number of elements for sorting is less than a certain number, we use *insertionSort ().* Note in the beginning of the class, there is a constant int named

CUTOFF which is initialled to 7. So when the number of elements for sorting is less than *CUTOFF*, choose *insertionSort ().*

Can you think of other ways to improve the performance?

Note: If you finished earlier, you can work on your assignment 2.