# Teacher’s Guide

# QuickSort

**OBJECTIVES:** The student, given a small list of unsorted data (nine elements), will trace the QuickSort algorithm through at least one level of recursive calls.

The student will state the order of QuickSort and explain how this was determined.

The student will count steps QuickSort uses to sort appropriate numbers of random integers.

The student will graph the data (steps versus number of elements) for the five sorting algorithms covered so far in this curriculum.

**ACTIVITIES/TIME:** Two Weeks

**MATERIALS:** Student Lesson AB26, *QuickSort*

Lab Assignment AB26.1, *quickSort*

Transparency AB26.1, *Action of quickSort*

Transparency AB26.2, *Log2N Steps*

Transparency AB26.3, *quickSort*

Worksheet AB26.1, *QuickSort1*

Worksheet AB26.2, *QuickSort2*

Handout AB26.1, *quickSort Method*

Teacher’s Guide AB26, *QuickSort*

Lab Assignment AB26.1, Answers – *Sorts.java, SortsCounting.java, QuickSort Results.doc*

Worksheet AB26.1, *Answer Sheet*

Worksheet AB26.2, *Answer Sheet*

**REFERENCES:** **Interactive Data Structure Visualizations**

<http://www.student.seas.gwu.edu/~idsv/idsv.html>

A site with Interactive Data Structure Visualizations for Quicksort.

**Animated Algorithms**

<http://oopweb.com/Algorithms/Documents/AnimatedAlgorithms/VolumeFrames.html>

A great collection of animated algorithms.

**INSTRUCTOR**

**NOTES:** Students will receive a copy of the QuickSort algorithm in the lesson and in Handout AB26.1, *quickSort Method*. Lab Assignment AB26.1, *quickSort* will have students add the appropriate counting steps for QuickSort to analyze its efficiency. The additional exercise of graphing is designed to reinforce the O(N \* log N) number of steps in this algorithm. Students should notice the dramatic improvement of QuickSort and MergeSort over the quadratic sorts.

Transparency AB26.1, *Action of quickSort*, is included. In this blackline master, a printout was created at the top of each function call and when the recursion was terminated by a list of one. This transparency shows the order of the recursive calls and how the quicksort algorithm proceeds. However, it might be best to present a quicksort of the nine elements by writing out all of the steps on a whiteboard, especially the movement of the g and h pointers.

The solution to Lab Assignment AB26.1 includes a graph in a Word document (*QuickSort Results*). The graph illustrates the differences in efficiency for the quicksort in this lab and the quadratic sorts and mergeSort from the labs for Lessons A17 & A18. The java files, *Sorts.java* and *SortsCounting.java*, contain all of these sorting algorithms as well as the code to count the number of steps used. The step-counting code is commented upon to explain exactly what steps are counted.

**WORKSHEET**

**NOTES:** Worksheet AB26.1, *QuickSort1* provides more basic practice and Worksheet AB26.2, *QuickSort2* provides extra practice.