



## AP Java Lab 3.4 - Final Sorts

These sorting algorithms are the some of the fastest sorting algorithms available. These algorithms will form the foundation of your project for this unit.

### Skills:

- **Declaring and manipulating arrays.**
- **Introducing quicksort, partition II, customSort**

### Class:

- **ArrayMethods4**

### Methods:

- **public static int partition(int[] list1, int front, int back)**
- **public static void quickSort(int[] list1, int front, int back)**
- **public static void customSort(//depends on the sort)**

### Your Mission:

*This partition method has different arguments than your previous lab. This will help us write the quicksort algorithm. The easy way to accomplish this method is to use Arrays.copyOfRange and pass your copies to the other partition method. However, this will cost time and space, and you want this partition algorithm to be as fast as possible to help quicksort. Choosing a clever pivot is another way to improve this algorithm.*

#### **public static int partition(int [] list1, int front, int back)**

*Returns:* void

*Side-effect:* list1 from list1[front] to list1[back] is partitioned on either side of a pivot.

*QuickSort recursively partition the array until we have found an array where front and back are equivalent. The recursion is similar to mergeSort because it divides and conquers the array. Use the partition method in this algorithm. Return the new position of the pivot.*

#### **public static void quickSort(int[] list1, int front, int back)**

*Returns:* void

*Side-effect:* list1 is sorted in ascending order

***customSort() //Choose a sorting algorithm we haven't covered, research and implement it in java. Be prepared to discuss in class.***