

## CSC 241

### Lab 10

**Sorts.java** is a test harness program for testing various sorting methods. The program includes a **swap** method that is used by all of the sorting methods to swap array elements.

- Modify the program so that after calling a sorting method the program prints out the number of swaps required to sort an array of 50 random integers.
- Test the modified program by running the **selectionSort**, **bubbleSort**, **shortBubble**, **insertionSort**, **mergeSort**, **heapSort**, and **quickSort**.
- Modify the program so that after calling a sorting method the program prints out the number of comparisons required to sort an array of 50 random integers.
- Test the modified program by running the **selectionSort**, **bubbleSort**, **shortBubble**, **insertionSort**, **mergeSort**, **heapSort**, and **quickSort**.
- In order to run the program just once, a **backupValues** array can be used in the **initValues** method. Use it to save all of the random numbers assigned to the **values** array. Then, create a **resetValues** method to reset the **values** array back to the original values by using the **backupValues** array each time you call one of the sorting methods. Of course, the backup array doesn't need to be reset for the first call to a sort—just the subsequent calls to the other sorts, because you need to start with the original unsorted array.

Here is a sample output:

Initial Array

The values array is:

```
88 49 69 19 03 54 57 83 42 48
92 72 78 10 08 13 46 29 73 90
30 44 80 74 66 60 79 36 05 63
09 97 62 60 21 51 63 83 87 22
08 18 48 41 38 22 58 99 19 94
```

0 swaps.

0 comparisons.

SelectionSort

The values array is:

```
03 05 08 08 09 10 13 18 19 19
21 22 22 29 30 36 38 41 42 44
46 48 48 49 51 54 57 58 60 60
62 63 63 66 69 72 73 74 78 79
80 83 83 87 88 90 92 94 97 99
```

49 swaps.

1225 comparisons.

BubbleSort

The values array is:

```
03 05 08 08 09 10 13 18 19 19
21 22 22 29 30 36 38 41 42 44
46 48 48 49 51 54 57 58 60 60
62 63 63 66 69 72 73 74 78 79
80 83 83 87 88 90 92 94 97 99
```

621 swaps.

1225 comparisons.

#### ShortBubble

The values array is:

03 05 08 08 09 10 13 18 19 19  
21 22 22 29 30 36 38 41 42 44  
46 48 48 49 51 54 57 58 60 60  
62 63 63 66 69 72 73 74 78 79  
80 83 83 87 88 90 92 94 97 99  
621 swaps.  
1215 comparisons.

#### InsertionSort

The values array is:

03 05 08 08 09 10 13 18 19 19  
21 22 22 29 30 36 38 41 42 44  
46 48 48 49 51 54 57 58 60 60  
62 63 63 66 69 72 73 74 78 79  
80 83 83 87 88 90 92 94 97 99  
621 swaps.  
667 comparisons.

#### MergeSort doesn't swap

The values array is:

03 05 08 08 09 10 13 18 19 19  
21 22 22 29 30 36 38 41 42 44  
46 48 48 49 51 54 57 58 60 60  
62 63 63 66 69 72 73 74 78 79  
80 83 83 87 88 90 92 94 97 99  
0 swaps.  
223 comparisons.

#### QuickSort

The values array is:

03 05 08 08 09 10 13 18 19 19  
21 22 22 29 30 36 38 41 42 44  
46 48 48 49 51 54 57 58 60 60  
62 63 63 66 69 72 73 74 78 79  
80 83 83 87 88 90 92 94 97 99  
73 swaps.  
271 comparisons.

#### HeapSort

The values array is:

03 05 08 08 09 10 13 18 19 19  
21 22 22 29 30 36 38 41 42 44  
46 48 48 49 51 54 57 58 60 60  
62 63 63 66 69 72 73 74 78 79  
80 83 83 87 88 90 92 94 97 99  
49 swaps.  
419 comparisons.