Emmit Parubrub
April 20th, 2017

Test Results:

All tests were made with a certain amount of integers and 3 arrays: sorted, random, and descending. Looking at the sorts, it seems like radix is the fastest sort.

hybridSort Results:
After hybridSort with array of 100 integers:
Total time without overhead: 0.33333333333333326 milliseconds

After hybridSort with array of 1,000 integers:
Total time without overhead: 1.3333333333333335 milliseconds

After hybridSort with array of 10,000 integers:
Total time without overhead: 29.166666666666668 milliseconds

After hybridSort with array of 100,000 integers:
Total time without overhead: 3105.1666666666665 milliseconds

randomizedQuickSort Results:
After randomizedQuickSort with array of 100 integers:
Total time without overhead: 0.666666666666667 milliseconds

After randomizedQuickSort with array of 1,000 integers:
Total time without overhead: 7.166666666666667 milliseconds

After randomizedQuickSort with array of 10,000 integers:
Total time without overhead: 205.58333333333334 milliseconds

After randomizedQuickSort with array of 100,000 integers:
Total time without overhead: 26795.166666666668 milliseconds

shakerSort Results:
After shakerSort with array of 100 integers:
Total time without overhead: 0.6666666666666667 milliseconds

After shakerSort with array of 1,000 integers:
Total time without overhead: 10.5 milliseconds

```
After shakerSort with array of 10,000 integers:
Total time without overhead: 521.6666666666666 milliseconds

After shakerSort with array of 100,000 integers:
Total time without overhead: 59829.833333333336 milliseconds



heapSort Results:
After heapSort with array of 100 integers:
Total time without overhead: 0.6666666666666667 milliseconds

After heapSort with array of 1,000 integers:
Total time without overhead: 19.5 milliseconds

After heapSort with array of 10,000 integers:
Total time without overhead: 622.0 milliseconds

After heapSort with array of 100,000 integers:
Total time without overhead: 67027.33333333333 milliseconds



combSort Results:
After combSort with array of 100 integers:
Total time without overhead: 1.0 milliseconds

After combSort with array of 1,000 integers:
Total time without overhead: 1.8333333333333335 milliseconds

After combSort with array of 10,000 integers:
Total time without overhead: 5.583333333333334 milliseconds

After combSort with array of 100,000 integers:
Total time without overhead: 40.5 milliseconds




iterativeMergeSort Results:
After iterativeMergeSort with array of 128 integers:
Total time without overhead: 0.3333333333333333 milliseconds
```

```
After iterativeMergeSort with array of 1,024 integers:
Total time without overhead: 2.1666666666666665 milliseconds

After iterativeMergeSort with array of 8,192 integers:
Total time without overhead: 7.083333333333334 milliseconds

After iterativeMergeSort with array of 131,072 integers:
Total time without overhead: 44.33333333333333 milliseconds



bucketSort Results:
After bucketSort with array of 100 integers:
Total time without overhead: 1.3333333333333333 milliseconds

After bucketSort with array of 1,000 integers:
Total time without overhead: 2.0 milliseconds

After bucketSort with array of 10,000 integers:
Total time without overhead: 6.666666666666667 milliseconds

After bucketSort with array of 100,000 integers:
Total time without overhead: 21.333333333333336 milliseconds



radixSort Results:
After radixSort with array of 100 integers:
Total time without overhead: 0.5 milliseconds

After radixSort with array of 1,000 integers:
Total time without overhead: 1.3333333333333335 milliseconds

After radixSort with array of 10,000 integers:
Total time without overhead: 2.3333333333333335 milliseconds

After radixSort with array of 100,000 integers:
Total time without overhead: 10.0 milliseconds
```