Group 7

Jaden & Harley

Our Code’s Final Times

This information is taken from scoreboard.txt in the “final” folder from the Sorting Competition Repository:

final1.txt: Median of 2715.0 milliseconds

final2.txt: Median of 196.0 milliseconds

Placed 7th Overall (8th in 1 and 7th in 2)

Code Description/Process

- Originally used comparison-based sorting
- Switched to bucket sort using quicksort for each bucket
- Swapped quicksort for radix sort in each bucket…

```
private static void bucketSort(String[] array, String target){
    SortingCompetitionComparator comparator = new SortingCompetitionComparator(target);
    int n = target.length();
    int len = array.length;

    int[] distances = new int[len];
    for (int i = 0; i < array.length; i++) {
        distances[i] = comparator.distanceToTarget(array[i]);
    }   //temporarily storing distances for use

    ArrayList<ArrayList<String>> buckets = new ArrayList<>(n+1);
    for (int i = 0; i <= n; i++) {
        buckets.add(new ArrayList<>());
    }   //make n buckets

    for (int i = 0; i < len; i++) {
        buckets.get(distances[i]).add(array[i]);
    } // adding distances to their buckets

    int index = 0;
    for (ArrayList<String> bucket : buckets) {
        int size = bucket.size();
        if (size > 1) {
            String[] bucketArray = bucket.toArray(new String[0]);
            radixSortBinary(bucketArray);
            for (int i = 0; i < size; i++) array[index++] = bucketArray[i];
        } else if (size == 1) {
            array[index++] = bucket.get(0);
        }
    }   //sort buckets using radix sort and add them to one big array
}
```

References

Bucket Sort: https://www.geeksforgeeks.org/dsa/bucket-sort-2/

Helped with creating buckets & concatenating buckets

Radix Sort: https://www.geeksforgeeks.org/dsa/radix-sort/

Just a general reference along with the textbook

Worst Case Running Time & Data Storage

- Worst Case Running Time: O(L*n)
  - L = number of digits (length of strings)
  - n = number of strings

- Longer binary strings = slower run time

- All data is stored in arrays

Improvements We Would Have Made

Changing the sorting method used within bucket sort would have made our program much more efficient

Insertion Sort

- Works better on smaller datasets, like the ones in our buckets

VS

Radix Sort

- Works better on larger datasets within a fixed range, not ideal for our buckets