

Group 1

Resa & Yuting



Performance

- **Sorting Times: first dataset**
 - Times: 1264, 1252, 1196
 - Median: 1252.0
 - Sorted Correctly
- **Sorting Times: second dataset**
 - Times: 271, 266, 268
 - Median: 268.0
 - Sorted Correctly
- **We placed 7th overall.**



How We Represented our Data: NumHolder

- **NumHolder (our data type) had 3 fields**
 - String original (original string)
 - Int num (original string converted to integer w/out “0.”)
 - Int modNum (Mod 10 of first four digits)

The first thing our algorithm did was loop through the given data and put it into a NumHolder array.



Sorting Algorithms

- **Counting Sort**

- We used a reverse counting sort on the modNum to sort it by the first four digits mod 10 from largest to smallest.

- **Quicksort**

- Then we used median quicksort on the the integer representation of the number. (We used the median of 3 or 9 based on the length of the section as our pivot. And we used a dual pivot.) We called this separately on each mod-number section.

- **Insertion Sort**

- We used insertion sort once the length of the recursion section was less than 7. (As part of the quicksort iteration).



Finishing Up

Once we had a properly sorted NumHolder array, we simply looped through it and copied the original string value back into the original array and returned it.



Technical Information

- **Running Time**

- Worst Case: $\Theta(n^2) + \Theta(n)$
- Expected Case: $\Theta(n \log_2 n) + \Theta(n)$
- Quicksort + Counting Sort
- The worst case will only happen when the pivots chosen are the min or max of the section everytime.

- **Memory Used**

- n
- Counting Sort (uses temporary arrays and makes a new result array)

- **Correctness Check**

- No problems found.

