Group 10: Daniel Frazier and Mitchell Finzel

SECOND PLACE

FIRST PLACE IN CLASS

SORTED CORRECTLY

Setup

- We used the original array of arrays for sorting
- We used an array of longs to store distances
- For the sorting algorithm we used a topdown merge sort implementation
  - Sorted the array of arrays as it sorted the array of distances

Efficiencies

- Running Time
  - Because of merge sort we had a worst case running time of $\Theta(n\lg n)$
    - The array is traversed once before sorting or $\Theta(n)$
- Memory
  - For our distances we had an extra array of longs n
  - Merge sort uses $O(n)$ extra memory

- For our algorithm we focused on minimizing cost of distance calculations
  - We found the line between the reference points that divided the grid by which reference point you are closer to.
    - This means you calculate the distance once for each point
  - The distance was then added to an array at the original indice
    - Points with a distance of zero were swapped to the front in both arrays
- Future work could focus on optimizing the sorting aspect