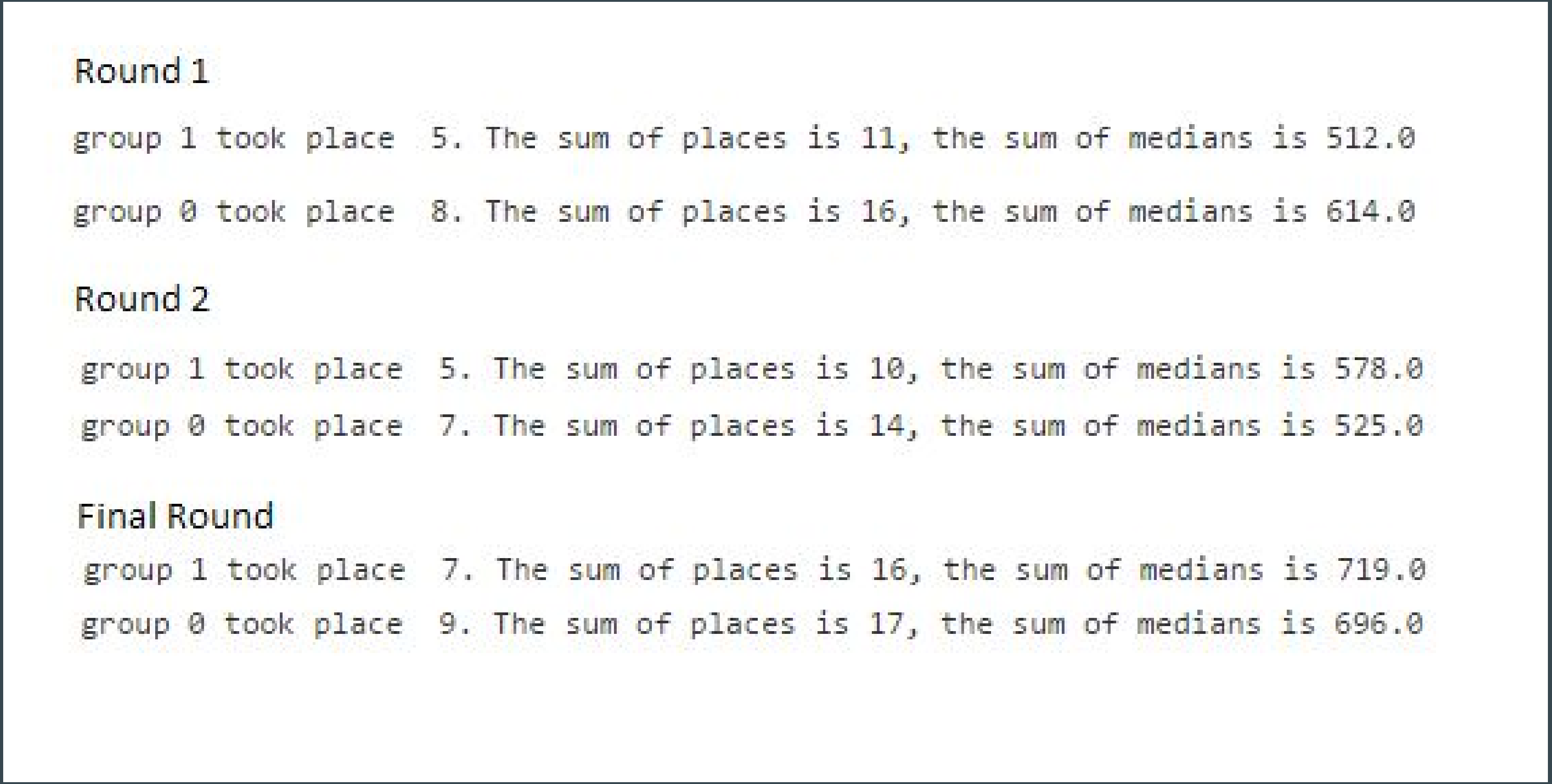
Group 1

Sam Miller & Partner
Sorting Competition 2016

Results

- 6th place overall; 5th place in class
- Correct Sorting
- Sum of places: 16
- Sum of Medians: 719.0



Round 1

```
group 1 took place  5. The sum of places is 11, the sum of medians is 512.0
```

```
group 0 took place  8. The sum of places is 16, the sum of medians is 614.0
```

Round 2

```
group 1 took place  5. The sum of places is 10, the sum of medians is 578.0
group 0 took place  7. The sum of places is 14, the sum of medians is 525.0
```

Final Round

```
group 1 took place  7. The sum of places is 16, the sum of medians is 719.0
group 0 took place  9. The sum of places is 17, the sum of medians is 696.0
```

Algorithm

- Hybrid merge sort that transitions to insertion sort when the subarrays reach a certain threshold
    - Used for its stable nature and good worst-case performance
    - Algorithm modified from classical implementation to enable use of insertion sort with a threshold tailored to the specific input data
- Uses int arrays and double int arrays to hold data
- No correctness concerns

Run Time

Run Time -> $\Theta(n \log n)$, unless $n < 7$, then it's $\Theta(n^2)$

(since 6^2 is a constant, as long as $n > 6$.)

Non-constant memory use -> n by virtue of mergesort.

What we tried

- Shell sort
- Quick sort
- Marking sequences of walks
- Introsort
- Just merge sort

What we could have improved

- Change how distance is calculated, make assumptions about where points can be in space
- Could have implemented a dynamic threshold that changes depending on characteristics of the input data