Group 8

Maggie Casale & Ben Simondet

Summary

- 11th Place overall
- 5th Place in our class
- The sum of places was 23
- The sum of medians was 2443.0
- Had correct sorting
- With 2,500,000 elements: 1930, 1893, 1914
- With 724,359 elements: 529, 530, 526

General Sorting

- One array of 10 arrayLists
- Each arrayList was used as a bucket
- Filling buckets
  - Ex) String = 0.012136527
    - mod = 4 % 10 = 4
    - String goes into bucket 4.
- Looping filled buckets
  - We go through each bucket, starting with the last (9) going backwards (to 0) and call Collections.sort on each arrayList.
  - The sorted bucket (arrayList) is then added back into the original array, placing elements starting at the beginning of the original array.

Efficiency

```
public static void sort(String[] arr){
        @SuppressWarnings("unchecked")
        List<String>[] buckets = (ArrayList<String>[])new ArrayList[10];
        int arrSize = arr.length;
        for(int k = 0; k < 10; k++){
                buckets[k] = new ArrayList<String>(arrSize/10);
        }
        String holder;
        for(int i = 0; i < arrSize; i++){                       O(n)
                holder = arr[i];
                buckets[getSumFirstFourDigits(holder)].add(holder);
        }
        int pos = 0;
        ArrayList<String> holderArr;
        for(int i = 9; i > -1; i--){
                holderArr = (ArrayList<String>) buckets[i];
                Collections.sort(holderArr);
                for(int j = 0; j < holderArr.size(); j++){      O(n log(n))
                        arr[pos++]=holderArr.get(j);            n/10 * 10 = O(n)
                }
        }
}                                                               +
                                                                ----------
                                                                Θ(3n log(n))
```

Extra

- Memory
  - Buckets extra memory: n
  - Collections.sort = TimSort: n
  - Total Extra Memory: 2n
- Things we would have done
  - Experimenting more with storing data in arrays vs arrayLists
  - Using charAt(i) to access integers for modulo 10 vs new Integer(s.substring(i, i+1))
  - Storing data as another type than string