Group 7

Jadyn S. and Isabelle H.

The Algorithm

Continued to use TimSort. We optimized both the counting binary ones and the lengthLongestRepeatedSubstring methods in the Helper class. For example, we managed to optimize lengthLongestRepeatedSubstring by minimizing the amount of *for* loops used.

We also tried bucket sort with the buckets being how many binary ones a number contained and then sorting the buckets by their longest repeated substring. But ran into some problems with it not sorting correctly so we decided to abort.

Score and Times

First Dataset: 10th place

- 3058
- 3117
- 3069

Second Dataset: 9th place

- 5338
- 5362
- 5292

Data stored in Integer Arrays

No correctness issues

Run Time

Timsort:

- Worst case: O(n *log*(n))
- Expected case (n *log*(n))

Our Comparator:

- Worst case: $O(m^2)$
- Expected case: $O(m^2)$

Worst / Expected case : $O(m^2 * n log(n))$

If we had more time, we probably would have looked further into bucket sort and found a way to implement it properly.