An analysis of O(n^2), O(n*lg(n)) and O(n) runtimes

For our third programming assignment, Dr. Corwin asked that we write a program to solve a simple problem - how many similar numbers are in two lists. To test this program, I wrote a simple script to generate {10, 100, 1000, 10000, 100000, 1000000 and 10000000} unique random integers, and give them to the c++ program to solve the problem. We were asked to do so in 3 unique ways:

1. With a nested loop - meaning that for each value in the first array, compare it to every value in the second array
    1. This algorithm runs in O(n^2) - it was quite apparent for lists of size greater than size 1000 that it would take much longer than the other solutions.
    2. This algorithm was unable to complete the given task in under 30 seconds for data sets greater than size 100,000.
2. With a sorted loop - Each list was sorted, and then by comparing values in linear time
    1. This algorithm runs in O(n * lg(n))
    2. This algorithm completed the task for each data set in under 2.5 seconds.
3. With a hash insert and hash find
    1. By doing two linear operations, this algorithm runs in O(n), ignoring constant multiples
    2. This algorithm completed data sets smaller than 1,000,000 incredibly fast, but due to the insane number of collisions provided my by pseudo-random number generator, ran at about the same speed as the second option for the data sets of 1,000,000 and 10,000,000 integers.

It is apparent that n^2 algorithms fall off fast when it comes to performance. They may be straightforward, but for large data problems, they are incredibly slow. With some simple arithmetic, you can see that for an n^2 algorithm to do 10,000,000 constant time comparisons, it would take 100,000,000,000,000 (100 Trillion) operations. Even a modern processor running at 2 GHz would take 50,000 seconds to do this - over 12 hours. And that's assuming that every processor cycle is spent comparing numbers. The n*lg(n) algorithm does this same data set in 2 seconds, approximately the same about of time as the O(n) algorithm. Provided is a screen shot of each algorithm's time and result for the 7 data sets.