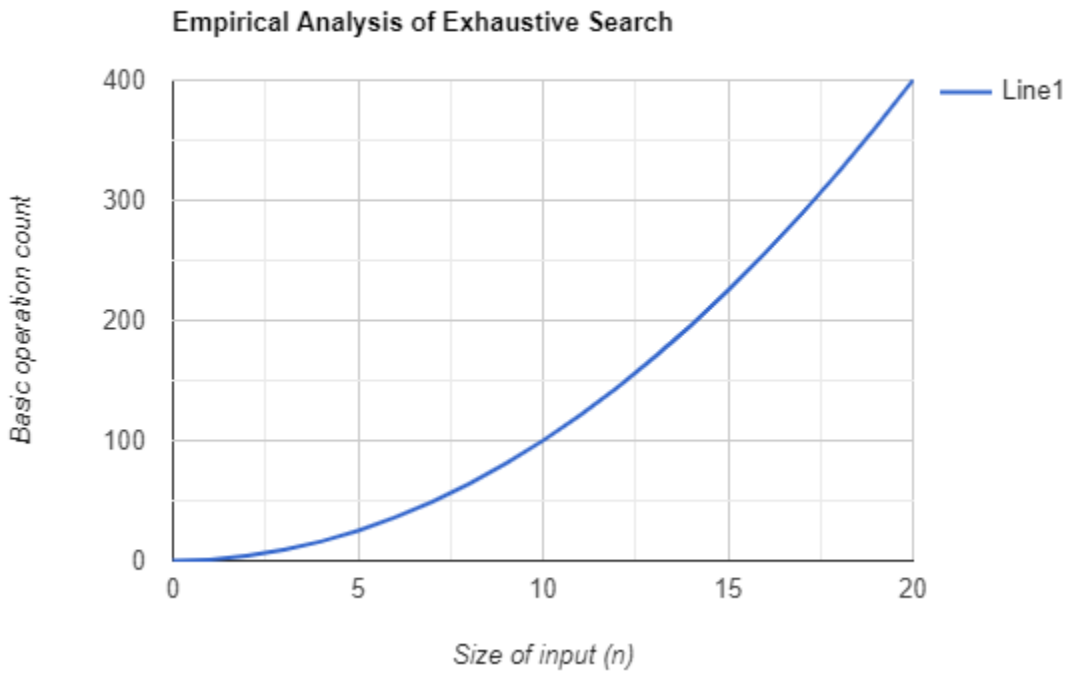
This exhaustive algorithm solves the closest pair problem, using “distance(list[i], list[j]) < smallest” (Line 110 of ClosestPair.java) as it’s basic operation. This basic operation runs n^2 times, where n is the length of the input array, because every index in the array is checked against every other index, even when it is redundant to do so. For example, given points 0 and 1, it will check the distance of: 0 -> 0, 0 -> 1, 1 -> 0, and 1 -> 1, even though 0 -> 1 would have been sufficient.

Empirically, we can test and see that it does indeed check exactly n^2 times, every time it is run, giving us a very simple graph for the amount of work done:



Comparing this to the divide and conquer method, we can see just how much more quickly this algorithm takes to check all of the possibilities. Even if we were to optimize this algorithm, it would still fall extremely short of divide and conquer. Even though this is far easier to write and to implement, the astronomical difference in computational work would make it worthwhile.