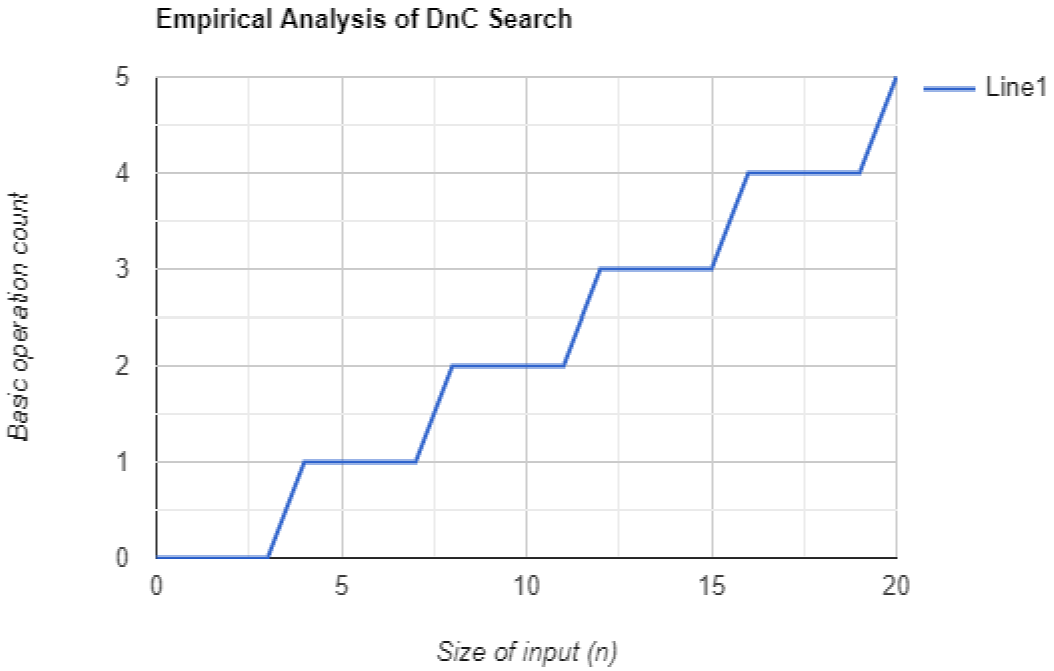
This algorithm calculates the problem recursively. This means that to calculate the work done, we must use a recursive formula. Given that the basic operation is recursively splitting the input in half, (Lines 149 & 150 of ClosestPairs.java) the work done by the algorithm can be approximated as n/4, where n is the length of the input array. This is shown to hold exactly true as for a while, but as n begins to grow, the amount of work will deviate. The graph shows that n/4 is accurate at least until n = 20, but at n = 10,000, the basic operation count is 2650. (about 6% higher than predicted)



Comparing this to the brute force method of exhaustively checking every possible combination, we can see that by n = 20, there is already a 20x performance increase due to the reduction in basic operations. This method takes more time to implement, but is absolutely justified by it's computational efficiency.