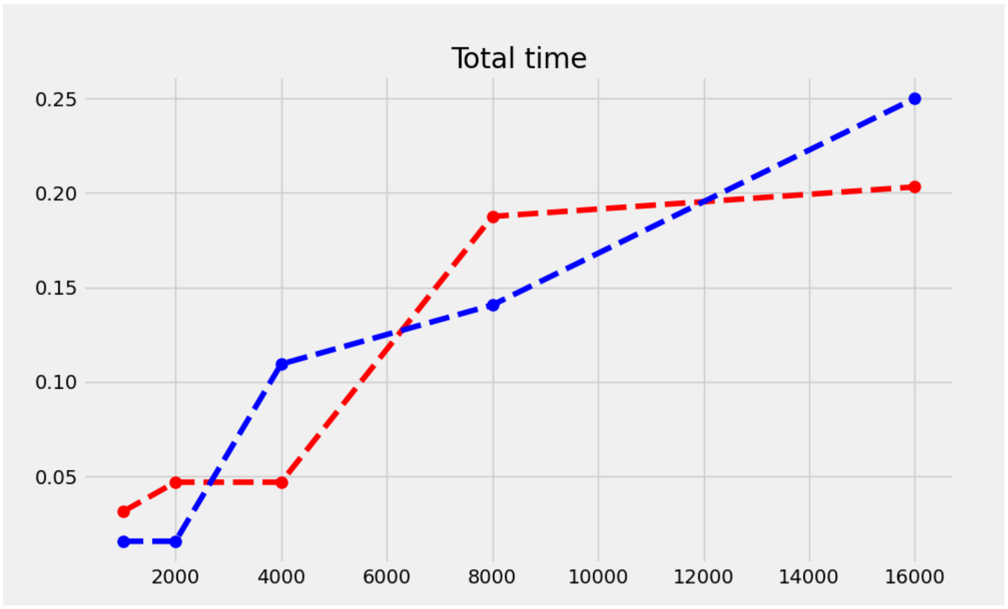
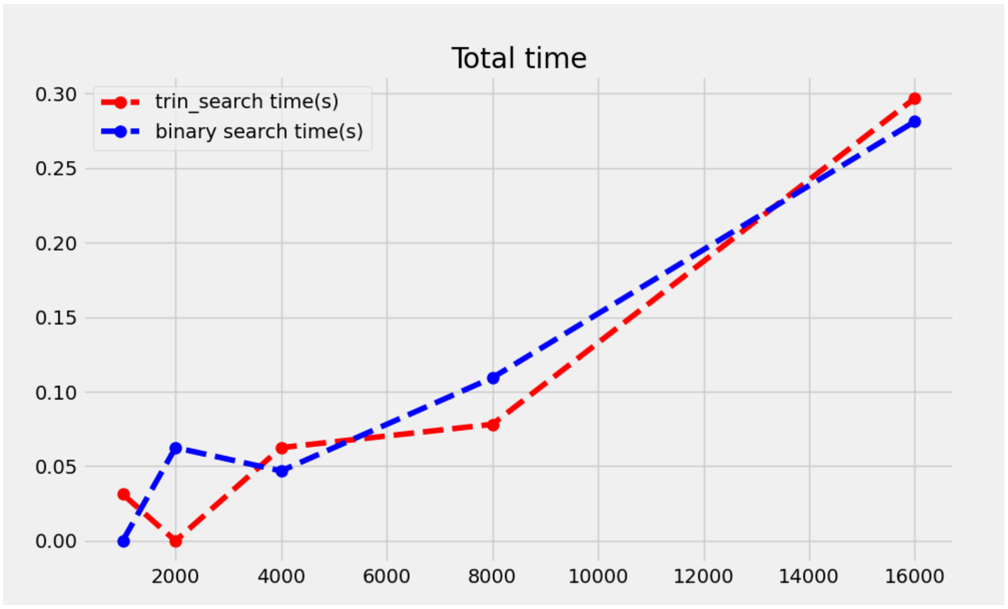
Project 1

First we observe what happens to the two algorithms in two different cases:

Experiment 1



Experiment 2



Question 1

For `BinarySearch` , the recurrence of the algorithm is $T(n) = T(\frac{n}{2})$, while for `TripleSearch` , the recurrence of the algorithm is $T(n) = T(\frac{n}{3})$, using Master's Thm we can obtain the time complexity of the algorithm as But the complexity of n only represents the execution of each line, and we cannot see this in time S. Moreover, the speed of each computer is different, which leads to a reduction in running time, so we cannot observe this in time alone, but if for the search we add a measure of then by using this value and the input size of the algorithm we can observe that the algorithm is $O(log(n))$.

Question 2

Looking at the times generated by the two algorithms for different input sizes, we can clearly observe that the difference between the times of `bin_search` and `trin_search` is not always kept within 10% in both cases, as can be seen in case 1 for $N = 4000$,I think that in the case of search, the time speed of `trin_search` and `bin_search` depends on many factors, the time complexity of the algorithm is just a mathematical analysis, the implementation of the algorithm and the efficiency of the language also determine the time of the algorithm, after many runs, I observed that for different N, it happens that `trin_search` is faster than `bin_search` or `bin_search` is faster than `trin_search` .