Experiment 1

n	Binary Search	Trinary Search
1000	16.974	18.256
2000	18.964	20.53
4000	20.9585	22.456
8000	22.9555	24.6785
16000	24.953875	26.81175

Experiment 2

n	Binary Search	Trinary Search
1000	19.954091816367267	21.968063872255488
2000	21.953046953046954	24.615384615384617
4000	23.952523738130935	26.358820589705147
8000	25.952261934516372	28.495876030992253
16000	27.952130983627047	30.990376202974627

1. At very large ‘x’-values (in this case ‘n’), in a logarithmic function, the increase in the y-axis (in this case the average comparisons per each n) becomes almost constant. In observing both experiments, trinary search’s and binary search’s average comparisons double as ‘n’ also doubles, thus, binary and trinary search both fall into the O(log n) complexity class.
2. In experiment 2, for n = 2000 the average amount of comparisons for binary search was less than 90% of the average amount of comparisons for trinary search. However, an interesting thing to note is that for experiment 1, for the same n value of 2000, binary search is not less that 90%.
3. From my experiments, it is very evident to conclude that overall binary search is preferable over trinary search. It makes sense, because there are less comparisons conducted in binary search than trinary search.

Kavin Zhu
20167040
“I confirm that this submission is my own work and is consistent with the Queen’s regulations on Academic Integrity”