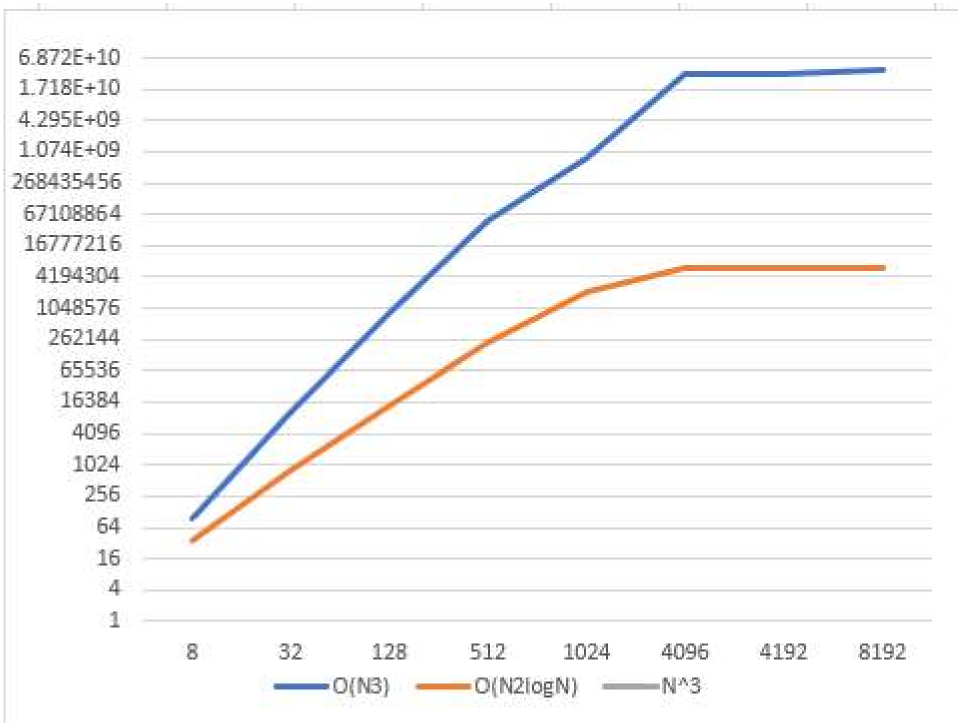
DATA STRUCTURES AND ALGORITHMS

HOMEWORK – 1

1. **3 Sum Algorithm**

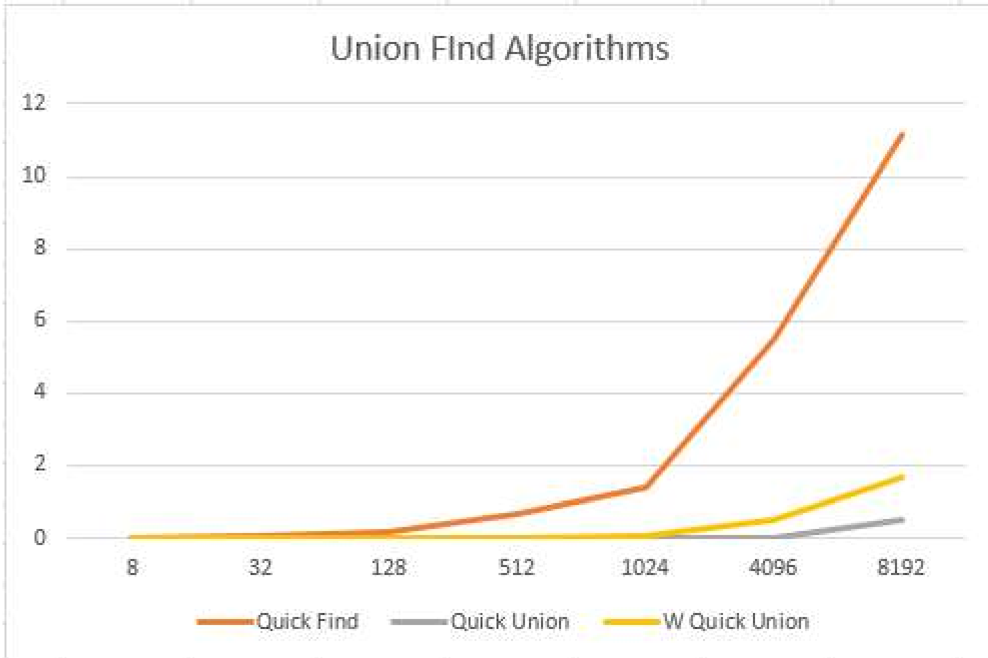
The graph for the brute force algorithm and binary search based 3 sum algorithm for different sizes is shown below:



From the graph, we can conclude that the line in blue(brute force) has a lot more instructions to be executed compared to the binary search-based algorithm. This proves that the binary search-based algorithm is more efficient than the brute force algorithm.

2. **Union Find Algorithm**

   The graphs for quick find, quick union and quick union with weighted balancing is shown below:



   From this graph, we can see that quick find and quick union have higher number of instructions to be executed compared to the weighted quick union, demonstrating that thee weighted quick union algorithm improves the performance of the quick union algorithm.

3. For the brute force algorithm in Q1,
   $f(N) = (N-1)(N-1) = (N-1)^3$ and $g(N) = N^3$

   For f(N) < cg(N):
   Where c = -1:

   N^3 – 3N^2 + 3N - 1 < -N^3
   2N^3 + 3N^2 - 3N + 1 < 0
   => (2N – 1)(N^2 – N + 1) > 0
   => **Nc > ½**

   For the binary search algorithm:
   (N – 1)^2(N/2 + N/4 + ….) < N^2logN

$N^2 – 2N + 1 < N^2$
$-2N + 1 < 0$
$2N – 1 > 0$
$N > ½$

For quick find:
$(M – 1) (N – 1) < MN$
$MN – M – N + 1 < MN$
$-M – N + 1 < 0$
$M + N – 1 > 0$

For M = N
$2N – 1 > 0$
$N > ½$

For quick union:

The procedure is similar to quick find.

For quick union with weighted balancing:

$(N – 1) + (M – 1)(M/2 + M/4 + ….) < N + M\log N$

$(N – 1) + (M – 1)\log N < N + M\log N$

$N + M\log N – 1 – \log N < N + M\log N$

$\log N + 1 > 0$

$\log N > - 1$

⇨ $N > 2^{(-1)}$
⇨ $N > ½$