

**Benchmarks:**

I have benchmarked both of my implementations of the convex hull algorithm, the divide and conquer version on input sizes of 0, 3, 5, 10, 100, 1,000, 10,000, and 20,000 5 times each and the naive version on input sizes of 0, 3, 5, 10, 30, 50, 100, 250, and 400, and then averaged the 5 runs to obtain a run time for each input size, and then I have plotted those run times onto a graph with a best fit curve and  $R^2$  value, the individual run times are listed in the tables below:

**DNC Algorithm:**

0:

Attempt	Run Time
1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
Average:	0.000

3:

Attempt	Run Time
1	0.001
2	0.000
3	0.000
4	0.000
5	0.000
Average:	0.0002

5:

Attempt	Run Time
1	0.001
2	0.001
3	0.000
4	0.000
5	0.001
Average:	0.0006

10:

Attempt	Run Time
1	0.001
2	0.000
3	0.001
4	0.001
5	0.001
Average:	0.0008

100:

Attempt	Run Time
1	0.003
2	0.003
3	0.003
4	0.004
5	0.003

Average:	0.0032
----------	--------

1,000:

Attempt	Run Time
1	0.047
2	0.043
3	0.046
4	0.044
5	0.045
Average:	0.045

10,000:

Attempt	Run Time
1	7.500
2	7.482
3	6.796
4	7.577
5	7.187
Average:	7.3084

20,000:

Attempt	Run Time
1	54.074
2	59.326
3	55.848

4	59.780
5	55.962
Average:	57.1312

### Naive Algorithm:

0:

Attempt	Run Time
1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
Average:	0.000

3:

Attempt	Run Time
1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
Average:	0.000

5:

Attempt	Run Time
---------	----------

1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
Average:	0.000

10:

Attempt	Run Time
1	0.000
2	0.001
3	0.001
4	0.001
5	0.001
Average:	0.0008

30:

Attempt	Run Time
1	0.035
2	0.034
3	0.032
4	0.033
5	0.033
Average:	0.0334

50:

Attempt	Run Time
1	0.174
2	0.170
3	0.171
4	0.164
5	0.172
Average:	0.1702

100:

Attempt	Run Time
1	1.462
2	1.480
3	1.496
4	1.401
5	1.430
Average:	1.4538

250:

Attempt	Run Time
1	23.226
2	23.915
3	23.285
4	24.111
5	24.301

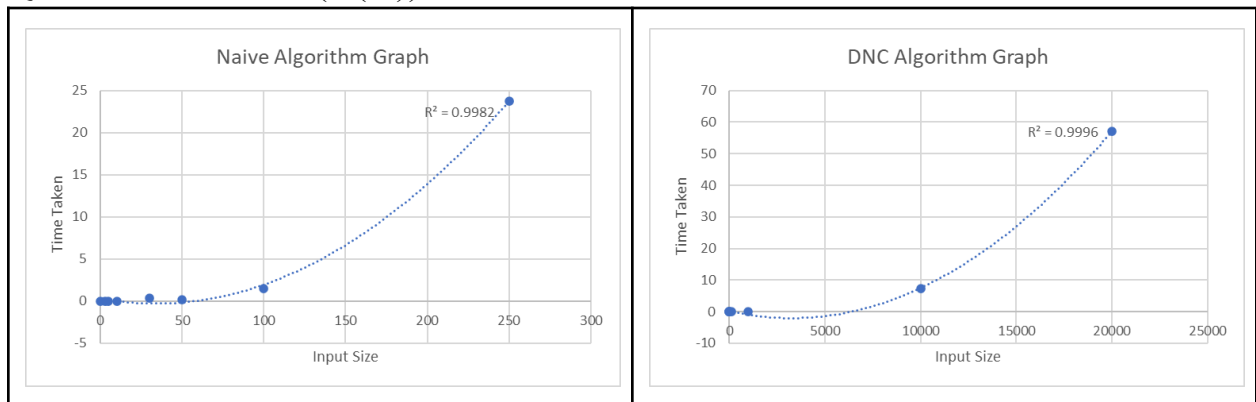
Average:	23.7676
----------	---------

400:

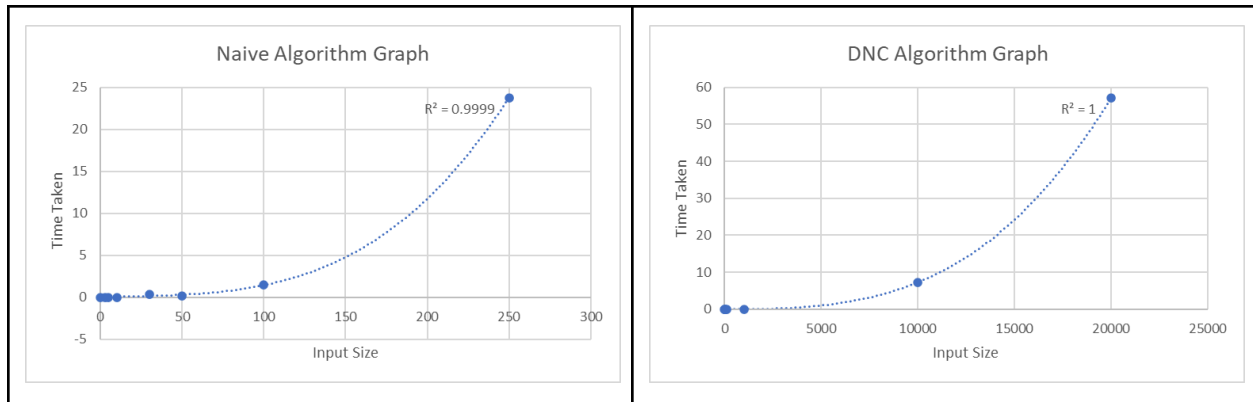
Attempt	Run Time
1	100.055
2	
3	
4	
5	
Average:	

## Best Fit Curve Graphs:

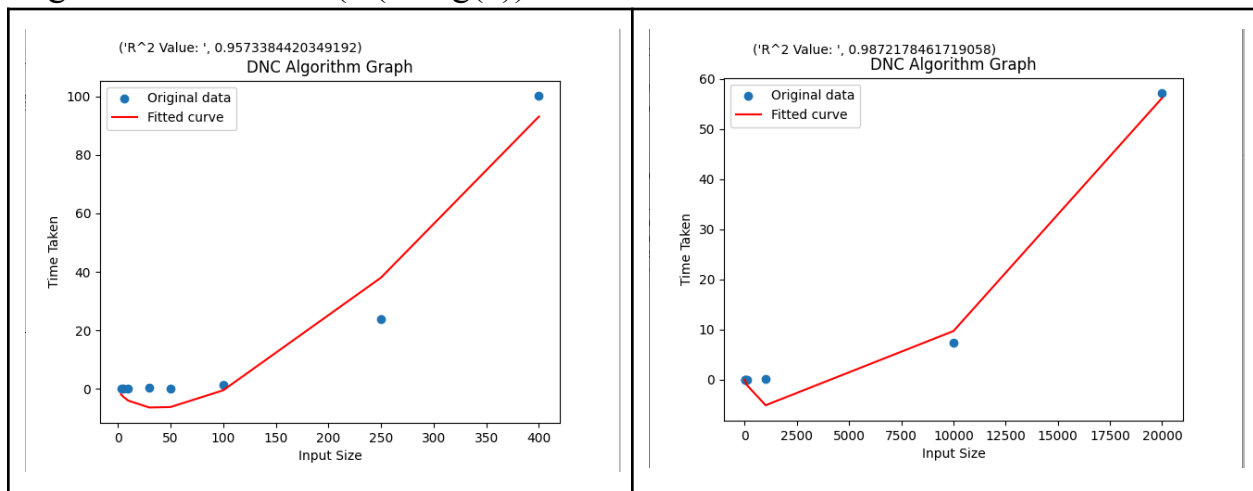
Quadratic Curve Fit ( $O(n^2)$ ):



Cubic Curve Fit ( $O(n^3)$ ):



### Log Linear Curve Fit ( $O(n \cdot \log(n))$ ):



If we look at these curve fits for the run times on a few different input sizes, we see a few different but very close  $R^2$  values, regarding the naive algorithm, we see the best  $R^2$  value given is for  $O(n^3)$  which is 0.9999. This coincides with my expected complexity of this algorithm which was also  $O(n^3)$ . Now if we look at the DnC algorithm, we see the same thing with the best  $R^2$  value being for , however I am still convinced this matches my expected complexity of the algorithm ( $n \cdot \log(n)$ ) as the  $R^2$  value for  $n \cdot \log(n)$  is still within acceptable values (close to 0.99) to consider it accurate.