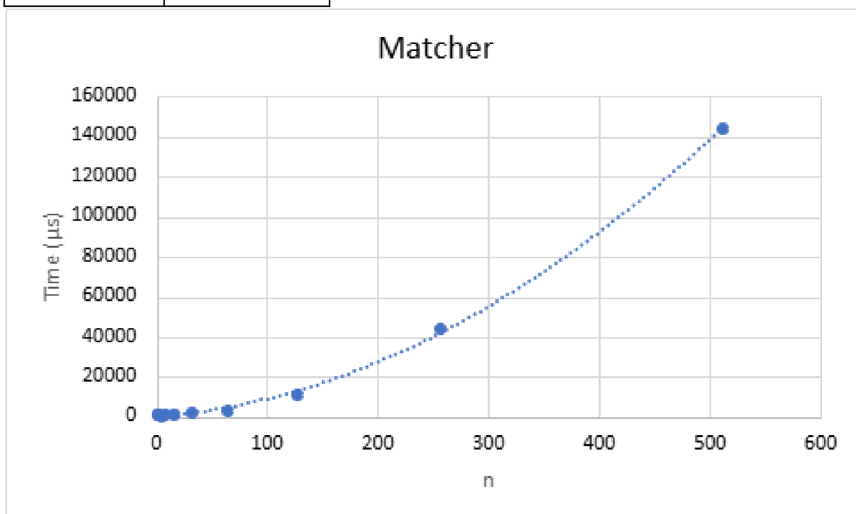
Programming Assignment 1: Part C

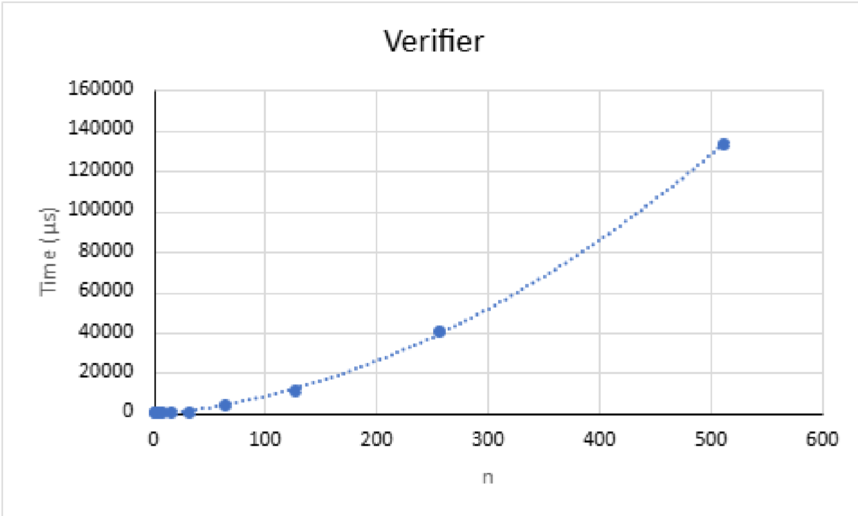
Matcher Table:

n	time
1	1002.8
2	997.4
4	0
8	1014
16	1012.6
32	1637.9
64	3096.6
128	11089.2



Verifier Table:

n	time
1	0
2	0
4	0
8	0
16	0
32	647.3
64	3591.7
128	11151.6
256	40514
512	133576.1



For Task C, we measured how long the matcher and verifier take to run as the number of hospitals and students increases. We tested values of n from 1 up to 512 and recorded the total execution time in microseconds for each run. The same input files were used for both the matcher and verifier so that the results would be comparable.

From the matcher graph, the running time clearly increases as n becomes larger. The growth is small for lower values of n, but it becomes much more noticeable after about n=128. The overall shape of the graph follows a polynomial trend. This matches the expected behavior of the Gale-Shapley matching algorithm, where in the worst case each hospital may need to propose to many students. Because of this, the time complexity of the matcher is $O(n^2)$.

The verifier graph shows a similar pattern. For small inputs, the runtime is very low, but it increases steadily as n grows. The verifier must read all preferences and check for both validity and stability, which involves examining many hospital-student pairs. This results in a worst-case time complexity of $O(n^2)$ as well. Although the verifier often runs faster than the matcher for smaller inputs, its runtime increases at a comparable rate for larger values of n.

Overall, the experimental results show that both the matcher and verifier scale in a polynomial form with respect to the input size. The observed trends in the graphs are consistent with the expected $O(n^2)$ time complexity for both algorithms.