1) The worst-case scenario for Quicksort time complexity is $O(n^2)$. Only the highest degree matters with time complexity and no contents are needed. $1/4 * n(n-1) = 1/4 * n^2 - n = O(n^2)$

3) While running the implementation between input sizes and testing the time. As the input size gets larger so does the time, however due to the setrecursionlimit() and O being (n^2) after the input size being greater than 2000, the program doesn't work or doesn't calculate the accurate time.

4) Data Points and Curve Fit: The red line shows the quadratic fit to these data points, while the blue dots show the actual execution times for various array sizes.Quicksort's temporal complexity analysis shows that, in the average situation, it is O(n log n), while in the worst scenario, it is O(n^2). Plotting shows that the execution time grows quadratically as the array size increases, which is consistent with quicksort's worst-case time complexity. Fit Quality: The execution time does, in fact, behave quadratically with the increase in array size, as the quadratic fit appears to closely resemble the real data points. Overall, the outcomes are consistent with the quicksort complexity study, which predicts that the execution time will grow quadratically with increasing array size.