1. As observed in class, quicksort's worst case complexity occurs when either the greatest or least value of an array is chosen as the "pivot." In both of these scenarios, quicksort operates with an array of n-1 elements and an array of 0 elements.

T(N) = the time needed to partition N elements + Time needed to sort (N-1) elements
= N + T(N-1)

This occurs recursively until the base case N=0 is reached.
Similarly,

T(N-2) = (N-2) + T(N-3)
T(N-3) = (N-3) + T(N-4)
And so on.

Therefore,
T(N) = N + (N-1) + (N-2) + … + 3 + 2
= [(N(N+1)) / 2 ] - 1
= O(N^2)

3. Implementation of quicksort on a number of inputs of increasing size

4. The observed results with appropriate interpolating functions do match the complexity analysis. Quadratic growth of the runtime can be observed as the size of the arrays increases, so does the runtime. The graph produced resembles a quadratic function, further supporting this.