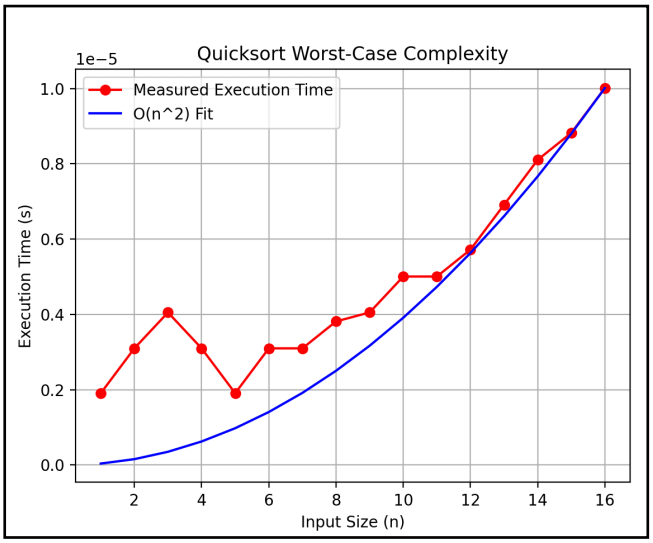
ENSF 338: Lab 3 Exercise 4

1. The worst-case time complexity of quicksort occurs when the pivot results in unbalanced partitions of the array. In other words, one side of the array is empty and the other side has all the elements except for the pivot. This case occurs when the input is already sorted, or if the pivot is the maximum or minimum value within the array. For the worst case of size n input, the recurrence relation can be illustrated through:
   $T(n) = T(n-1) + O(n)$
   Deriving the worst case time complexity using this formula:
   $T(n) = T(n-1) + n$
   $= T(n-2) + (n-1) + n$
   $= T(n-3) + (n-2) + (n-1) + n$
   $= 1 + 2 + 3 + \ldots + (n-1) + n$
   $T(n) = O(n^2)$

2. A vector of 16 elements that shows worst case complexity can be shown through:
   [9, 8, 7, 6, 5, 4, 3, 2, 1, 10, 20, 30, 40, 50, 60, 70]
   A. Choose pivot = 9:
   Partition: [9 | 8, 7, 6, 5, 4, 3, 2, 1, 10, 20, 30, 40, 50, 60, 70]. This causes the left side to be empty, whereas the right side contains all the other elements.
   B. Next Call: after applying the quicksort to the right side, choose pivot = 8
   Partition: [8 | 7, 6, 5, 4, 3, 2, 1, 10, 20, 30, 40, 50, 60, 70]. This causes the left side to once again being empty, and the right side with all the elements.
   C. Next call: after applying the quicksort to the right side, choose pivot = 7
   Partition: [7 | 6, 5, 4, 3, 2, 1, 10, 20, 30, 40, 50, 60, 70]. Once again, this causes the left side to once again being empty, and the right side with all the elements.
   D. This process is continued until the last element is reached.
   E. Final sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70]. Overall, this example of a worst-cast complexity for quicksort. As computed from the previous question, this leads to $O(n^2)$. As shown through the calculations from this example, these unbalanced partitions are created consistently by choosing the first element of the pivot, hence why it has to be the worst case.
3. (The code for part 3 is attached as a .py file)
4. 

The following plot shows that the measured execution time (indicated by the red line) closely follows the $O(n^2)$ trendline (blue line). This shows the quicksort's quadratic behaviour when using the worst case time complexity. Even though there are minor fluctuations because of system-level variations, the trend line overall matches with the analysis shown above. Other optimizations, such as randomized pivot selection could potentially improve the performance.