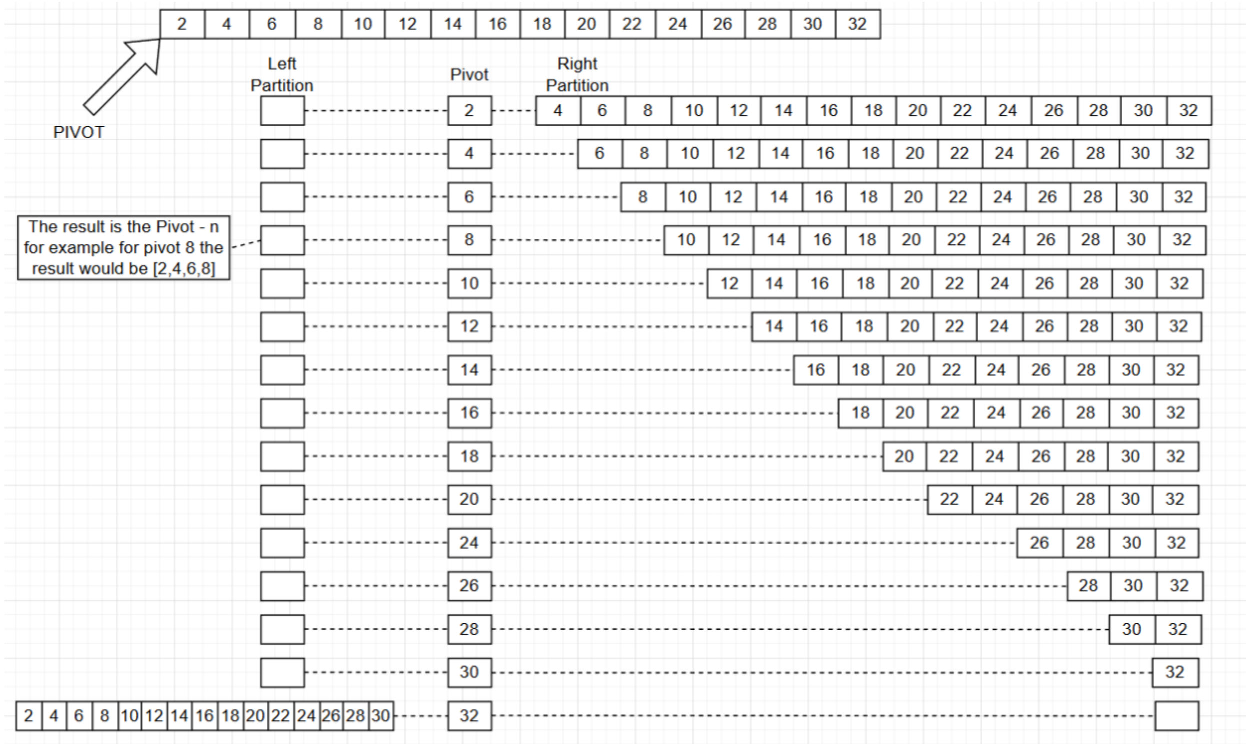
Exercise 4: More Complexity analysis

1.) The worst case for Quick sort $O(n^2)$ occurs when the largest or smallest element is always chosen for the pivot point. This results in a sub-array of size 0 and another with a size of n-1.

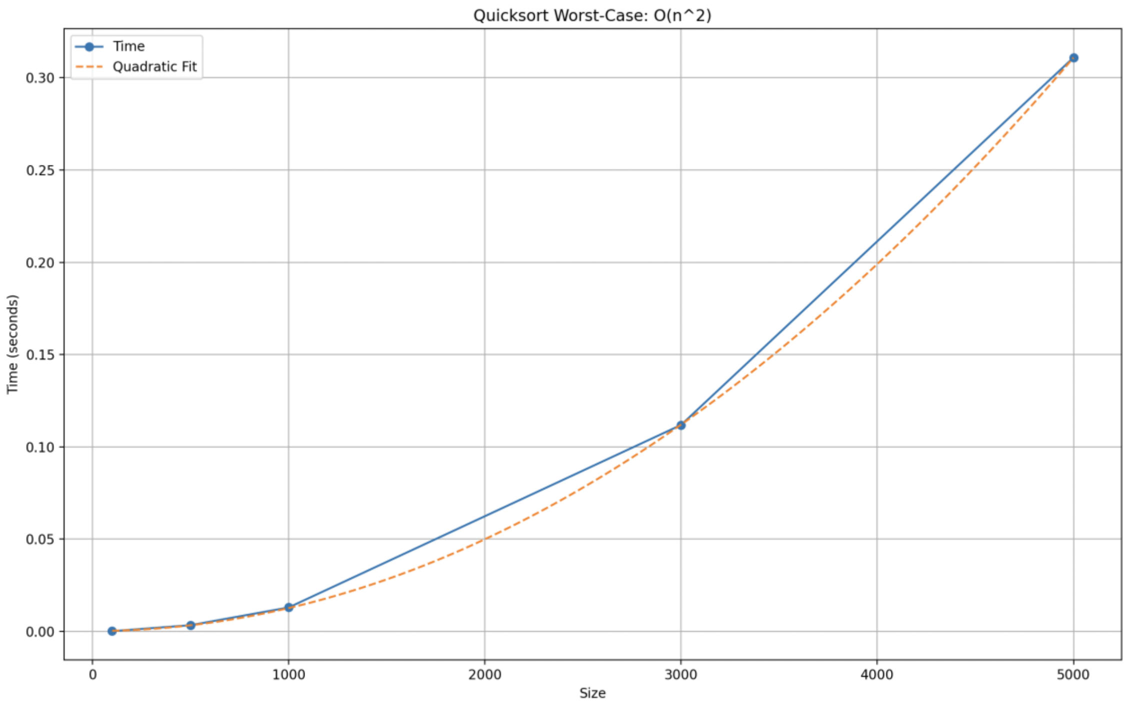
- $T(n) = T(n-1) + O(n)$
- Expand until base case is reached:
- $T(n-1) = T(n-2) + O(n-1)$
- $T(n-2) = T(n-3) + O(n-2)$
- $T(2) = T(1) + O(2)$
- Substitute the expansion back in:
- $T(n) = T(n-1) + O(n)$
- $T(n) = [T(n-2) + \ O(n-1)] + O(n)$
- $T(n) = [T(n-3) + O(n-2)] + O(n-1) + O(n)$
- $T(n) = T(1) + O(2) + O(3) + \cdots + O(n-1) + O(n)$
- Simplifying:
- $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$
- $O(\ 1 + 2 + 3 + \cdots + n\ \ ) = O\left(\frac{n(n+1)}{2}\right) = O(n^2)$
- Finally:
- $T(n) = T(1) + O(n^2)$
- $T(n) = O(n^2)$

2.)



2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32
PIVOT
Left Partition
Pivot
Right Partition
The result is the Pivot - n for example for pivot 8 the result would be [2,4,6,8]
2 | 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32
4 | 6 8 10 12 14 16 18 20 22 24 26 28 30 32
6 | 8 10 12 14 16 18 20 22 24 26 28 30 32
8 | 10 12 14 16 18 20 22 24 26 28 30 32
10 | 12 14 16 18 20 22 24 26 28 30 32
12 | 14 16 18 20 22 24 26 28 30 32
14 | 16 18 20 22 24 26 28 30 32
16 | 18 20 22 24 26 28 30 32
18 | 20 22 24 26 28 30 32
20 | 22 24 26 28 30 32
24 | 26 28 30 32
26 | 28 30 32
28 | 30 32
30 | 32
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 | 32

4.)



Since Quicksort's worst-case complexity is O(n^2) a quadratic interpolation function was selected to reflect this. Based on the graph this matches our complexity analysis because the time closely follow the quadratic fit. For n < 1000 the execution time is very low, and the deviation between the actual execution time and O(n^2) is minimal. As the size increases to n > 3000 the execution time follows a clearer quadratic growth pattern which is a characteristic of O(n^2). This validates the expectation that Quicksort degrades to O(n^2) when a bad pivot point is chosen, such as the first element.