Exercise #4 - More Complexity Analysis

1.
Start with:
T(n) = T(0) + T(n-1) + cn
T(n-1) = T(0) + T(n-2) + c(n-1)

Substitute T(n-1) into equation 1:
T(n) = 2T(0) + T(n-2) + c(n-1) + cn

Substitute T(n-2) into equation 3:
T(n) = 3T(0) + T(n-3) + c(n-2) + c(n-1) + cn

Continue this process until you reach the base case:
T(n) = nT(0) + c(n-1) + c(n-2) + ... + 2c + cn

Simplify
T(n) = O(n^2)

2.
The Initial Vector:
[16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

It generates a random permutation of the vector:
[10, 1, 6, 2, 12, 9, 15, 8, 3, 4, 7, 5, 14, 11, 13, 16]

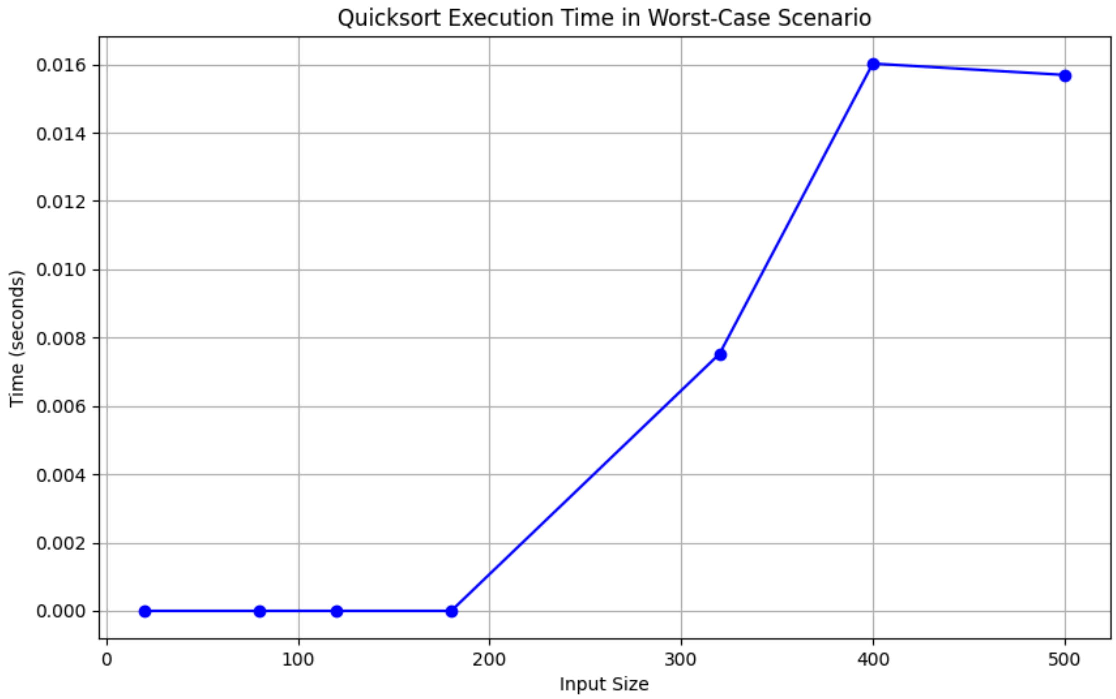
This permutation is not sorted…

It Generate another random permutation:
[3, 6, 11, 15, 1, 8, 14, 5, 7, 2, 4, 10, 9, 13, 12, 16]

Still not sorted.

It then repeats the process until it happens to generate a sorted permutation:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

4.



Yes, the results generally align with the expected complexities. The worst-case scenario exhibits a quadratic relationship, confirming the O(n^2) complexity, while the average-case scenario demonstrates a smoother growth consistent with O(n log n). The interpolating function aids in visualizing trends between measured points.