3. Mathematically derive the average runtime complexity of the non-random pivot version of quicksort.

To derive the average runtime complexity of the non-random pivot version of quicksort, we need to analyze the algorithm's behavior at each level of recursion. In this version, we typically select the pivot as the middle element of the array.

Let's assume we have an array of size n that we want to sort using quicksort. At each level of recursion, we partition the array around a pivot element and then recursively sort the two subarrays formed by partition. The partitioning step takes O(n) time, and we assume that the pivot selection also takes O(1) time.

At each step of Quicksort we choose a pivot element and partition the array into elements ~~similar~~ smaller than the pivot and elements larger than the pivot. This partitioning process takes O(n) time as each element needs to be compared to the pivot one.

After partitioning we recursively sort the subarrays on either side of pivot.

Assuming that the pivot is chosen randomly on average, we expect the subarrays to