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 the partition. The partitioning step takes $O(n)$ time, and we assume that the pivot selection (choosing the middle element) also takes $O(1)$ time.

In the best-case scenario, the pivot divides the array into two equal parts, resulting in $O(\log n)$ levels of recursion, each taking

$O(n)$ time for partitioning. Therefore, the best-case time complexity is $O(n \log n)$.

In the worst-case scenario, the pivot is either the smallest or largest element in the array, resulting in only one subarray being smaller than the original array by 1 element and the other subarray being empty. This leads to $O(n^2)$ time complexity, as each level of recursion only reduces the size of one subarray by 1 element.

To derive the average-case time complexity, we consider the average behavior over all possible pivot selections. If we assume that the pivot is chosen uniformly at random from the array, the probability of selecting any particular element as the pivot is $1/n$. Therefore, the probability that the pivot divides the array into two subarrays of size

k and $n-k-1$ (for some k) is $2/\mathrm{n}$

Let $T(n)$ denote the average-case time complexity of quicksort for an array of size

n. We can express $T(n)$ as:

$T(n)=O(n)+1/\mathrm{n}\sum_{k=0}^{n-1} [T(k)+T(n-k-1)]$

Here, the first term $O(n)$ represents the partitioning step, and the sum represents the recursive calls on the two subarrays. The sum is taken over all possible sizes

k of the left subarray.

Solving this recurrence relation is non-trivial and involves techniques such as the Master theorem or substitution method. The average-case time complexity of the non-random pivot version of quicksort is approximately

$O(n \log n)$ when the pivot is chosen as the middle element.