To derive the average runtime complexity of the non-random pivot version of quicksort, let's consider the following:

In the non-random pivot version of quicksort, we typically choose the pivot element as the last element of the array. Then, we partition the array into 2 sub-arrays such that all elements less than or equal to the pivot are on the left side, and all elements greater than the pivot are on the right side. After partitioning, we recursively apply the same process to the left and right sub-arrays.

Let's denote:

- n as the no. of elements in the array.
- $T(n)$ as the average runtime complexity of quicksort for an array of size n.

The average runtime complexity of quicksort can be expressed as a recurrence relation:

$T(n)$ = Time to partition + Time to sort left sub-array + Time to sort right sub-array.

The time to partition the array depends on the size of the array and the partitioning process. Assuming that the partitioning process takes $O(n)$ time, we have:

Time to partition = $O(n)$

The time to sort the left and right sub-arrays depends on the sizes of those sub-arrays. Since quicksort divides the array into 2 parts, each approximately half the size of the original array in the average case, we can write:

Time to sort left sub-array $= T\left(\frac{n}{2}\right)$

Time to sort right sub-array $= T\left(\frac{n}{2}\right)$

$\therefore T(n) = O(n) + 2 \cdot T(n/2)$

Now, we can use the Master Theorem to solve this recurrence relation.

The master theorem states that if $T(n) = aT(n/b) + f(n)$, where $a \geq 1$, $b > 1$, and $f(n)$ is an asymptotically positive function, then:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$

2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$ for some constant $k \geq 0$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.

3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af\left(\frac{n}{b}\right) \leq kf(n)$ for some constant $k < 1$ and sufficiently large n, then $T(n) = \Theta(f(n))$.

In our case, $a = 2$, $b = 2$, and $f(n) = O(n)$
we have $\log_b a = \log_2 2 = 1$.

Sence $f(n) = O(n) = O(n^{\log_b a - \epsilon})$ for $\epsilon > 0$ (with $\epsilon = 0.5$), we can apply case 1 of the master theorem.

$\therefore$ the average runtime complexity of the non-random pivot version of quicksort is $\Theta(n \log n)$.