Before diving into the parallelization of the quicksort algorithm, it is first necessary to briefly describe its serial version in order to have a baseline for comparison.
Given an unsorted input array $\mathbf{X}$ of n elements, the serial quicksort algorithm sorts the values of the array according to the following procedure:

1. choose a pivot value τ from the array $\mathbf{X}$;
2. partition the array $\mathbf{X}$ into two sub-arrays, one containing only elements smaller than the pivot, $\mathbf{X}_{<\tau}$, the other containing elements greater or equal to the pivot, $\mathbf{X}_{\geq\tau}$;
3. recursively sort the two sub-arrays $\mathbf{X}_{<\tau}$ and $\mathbf{X}_{\geq\tau}$ until the base case of having a sub-array of size 2;

The final sorted array $\mathbf{X}$ is then obtained by concatenatenation of the sorted sub-arrays. This sorting algorithm offers an average time complexity of $\mathcal{O}(n \log n)$, with the worst case scenario complexity being $\mathcal{O}(n^2)$ [**?**].