i] No, we can't decide the sequence of functions executed from gprof. In the output.txt file we can observe that functions executed from gprof have exactly the same sequence as it was in the code i.e. time required for swap() is given first then for partition() and finally quicksort().

ii] Generally quick sort is faster algorithm because it uses divide and conquer technique which roughly divides the list in half after each recursion call.

For a Quick Sort algorithm,

Best case is if the partitioning is in the middle always. The time complexity for this best case is O(nlogn) where log is of base 2.

Worst case is if the list is already sorted in ascending or descending order. The time complexity for this worst case $O(n^2)$.

Case	Time Complexity
Best Case	O(n*logn)
Average Case	O(n*logn)
Worst Case	$O(n^2)$

In the output.txt file we can see that for n=10 i.e. for an array of size 10, during the whole iteration the most called function is swap. It is called 54 times for 45 comparisons. Then the partition is called 9 times and quicksort once. The running time for this value is so negligible that it is showing 0.0 in the output.

As compared to this for n=10000, we can observe that there is non-zero runtime. So, for higher values of n, the time required for computation is also higher.