Quick Sort Algorithm

1. select a pivot element
   - first element
   - last element
   - middle element
   - random element
2. partition:
   - all elements less than the pivot go to its left.
   - all the elements greater than the pivot to its right
   - so now we can say that pivot kept on its correct position.
3. Sorting
   - sort them by swapping

Complexity:

Case	Time Complexity
Best Case	O (n log n)
Average	O (n log n)
Worst Case	O (n^2)

Worst case occurs when the pivot is always smallest or largest. (already sorted array in that you are choosing first / last element
as pivot)

Space Complexity:
O (log n) for recursive stack (in place sort, no extra array is used)

Sliding Window Approach:

Usecase:
maximum sum of sub array for size k
longest sub string with no repeating characters
minimum window to satisfy the condition