Adaptive median Filter Algorithm:

As shown in **Figure1** count sort is better than quick sort after a specific window size.

Why?

- Count sort algorithm is big O(n)
  as it counts frequency of every different number and then starting from the smallest number it repeats it in the array according to its frequency.
- Quick sort algorithm is big O(nlogn)
  as it chooses a pivot and put it in his right place then sort the subarray before the pivot and then the subarray after the pivot.
- But, in small window size case:
  Looping the 256 values of pixels is taking much time the normal quick sort of the small size of the array of the small window.

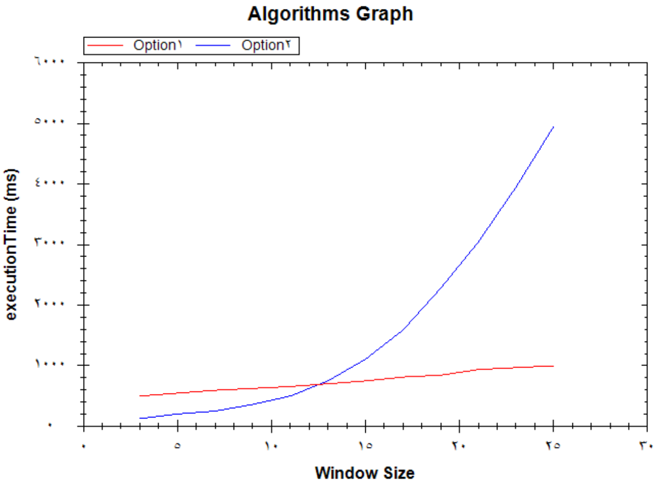

Figure 1

Option1: Count Sort

Option2: Quick Sort

Alpha trim filter:

As shown in **Figure2** count sort is better than selecting the smallest\largest kth elements.

Why?

- Count sort as shown before is big O(n).
- Selecting the smallest\largest kth elements is in average is o(n) as it just after partitioning the array it works on one subarray but what if the array is almost sorted?
  It can take almost $o(n^2)$ time in this case.

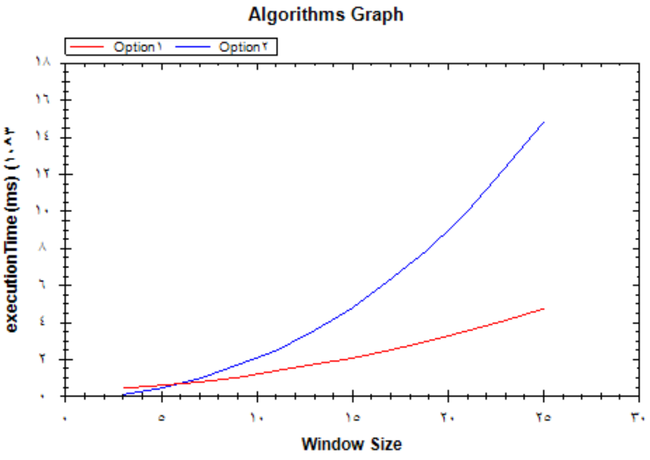

Figure 2

Option1: Count Sort

Option2: Select kth smallest and kth largest elements