Summary report on quick sort and modified quick sort

QuickSort is an algorithm based on the divide and conquer strategy to recursively sort elements by choosing a pivot element and using it to split the numbers in two arrays, one containing numbers smaller than the pivot and other containing numbers larger than the pivot.
Time taken by QuickSort can be written as follows:

$T(n) = T(k) + T(n-k-1) + (n)$

Run time : Best case - $O(n \log_2 n)$
Worst-case - $O(n^2)$

The best case occurs when the pivot we pick happens to divide the array into two exactly equal parts, in every step. Worst case can occur when the pivot choosen is the smallest or largest element of the set. By choosing the last or first element as a pivot, the complexity can reach the worst case in case of a sorted input.

There are a number of ways to improve the worst case performance of quicksort. One such method is by choosing the pivot in a different way(Median of three method) rather than the last or first element of the array.

The median of three method :
Choosing the pivot by this method helps to make quicksort perform in O (n log n) time even for ordered or reverse-sorted input by estimating an optimal pivot.

Observations of the implementation results :
My choice Quicksort performs better incomparasion to normal quicksort in terms of CPU run time. As the data size increases the difference of CPU runtime is seen to be increased. My choice quicksort performs better for sorted and reverse sorted data.