Comparision between std::sort and quicksort by Minh Tran ID:40122374

In theory

The C++ Standard Library sorting function std::sort() is implemented by using the sorting algorithm called IntroSort.

> IntroSort is a hybrid sorting algorithm that begins a quicksort , switches to heapsort when the recursion depth exceeds a level based on the number of elements being sorted and it switches to insertion sort when the number of elements is below some thresholds.

> IntroSort
>
> - Worst-case performance: O(n * log(n))
> - Average-case performance: O(n * log(n))

while

> QuickSort
>
> - Worst-case performance: O(n * n)
> - Average-case performance: O(n * log(n))

So, in theory, std::sort (IntroSort) should be able to beat QuickSort

In reality

The code for both sorting algorithms is included in the zip file.

Each sorting algorithm was tested with a vector of 100000 32-bit integers for 10 times

Using QuickSort:

QuickSort	time in milliseconds
first run	37 milliseconds
second run	51 milliseconds
third run	38 milliseconds
fourth run	39 milliseconds
fifth run	39 milliseconds

QuickSort	time in milliseconds
sixth run	58 milliseconds
seventh run	54 milliseconds
eighth run	35 milliseconds
ninth run	44 milliseconds
tenth run	56 milliseconds

Average sorting time : 45.1 milliseconds.

Using std::sort:

std::sort	time in milliseconds
first run	53 milliseconds
second run	38 milliseconds
third run	30 milliseconds
fourth run	29 milliseconds
fifth run	38 milliseconds
sixth run	51 milliseconds
seventh run	33 milliseconds
eighth run	33 milliseconds
ninth run	32 milliseconds
tenth run	34 milliseconds

Average sorting time: 43.8 milliseconds.

Conclusion

Std::sort consistently beats QuickSort many times with the shortest shorting time at 29 milliseconds and slightly better average sorting time at 43.8 milliseconds