

Comparison

What are the differences between the three algorithms? For each algorithm, mention a situation where it has an advantage over the other two.

Time complexity	Best case	Average case	Worst case
Insertionsort	$O(n)$	$O(n^2)$	$O(n^2)$
Quicksort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$
Heapsort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$

Testing with function rangen with different values of the input n, to see how long each algorithm takes.

Test with n = 100

Insertionsort	Quicksort	Heapsort
0.0006861686706542969	0.0002701282501220703	0.0006749629974365234

Test with n=1000

Insertionsort	Quicksort	Heapsort
0.08333492279052734	0.0021657943725585938	0.006887912750244141

Test with n= 10 000

Insertionsort	Quicksort	Heapsort
8.958541870117188	0.030632972717285156	0.0913851261138916

Test with n= 25 000

Insertionsort	Quicksort	Heapsort
58.88928008079529	0.08368992805480957	0.2504119873046875

The insertion sort has a complexity of $O(n^2)$. Although it has the same complexity, the insertion sort is a little over twice as efficient as the bubble sort.

Heapsort is the slowest of the $O(n \log n)$ sorting algorithms but unlike quick sort it does not require massive recursion or multiple arrays to work.

The quick sort is a divide-and-conquer, recursive sort. The efficiency of the algorithm is impacted by which element is chosen as the pivot point. The worst-case efficiency of the quick sort is $O(n^2)$ when the list is sorted and left most element is chosen as the pivot. As long as the pivot point is chosen randomly, the quick sort has an algorithmic complexity of $O(n \log n)$.

The quick sort doesn't do unnecessary swaps since it is time consuming, meaning that if the data is ordered quicksort will swap hardly anything. This is significantly different to heapsort, even if its data would already be sorted heapsort will still swap 100% of the elements to order the array.

So generally speaking, the efficiency of these different algorithms depends on the input, for example the size and whether it is sorted or not. The displayed tables above concretely visualizes the differences timewise while executing their operations.