CMSC 251:Spring 2002 Don Perlis

Runtimes and other results for various comparison-based sorting algorithms

	Worst Case	Average Case	In place	Locality
Insertion Sort	$\frac{n^2}{2}$	$\frac{n^2}{4}$	Yes	Yes
Merge Sort*	$n \lg n$	$n \lg n$	No**	Yes
Heap Sort	$2n \lg n$	$2n \lg n$ ***	Yes	No
Quick Sort	$\frac{n^2}{2}$	$\sim 1.39 n \lg n$	Yes	Yes

*For a Merge procedure that always does exactly $n-1$ comparisons on each call with n items.

**Can be done in place, but not practical.

***Hard to prove.

The above expressions for runtime give the highest-order terms only, but with the correct coefficients.

This does not take account of other time-consuming aspects of the algorithms, especially data movement. But as a general rule in comparison-based algorithms, there are at most no more data movements than there are comparisons (since data is generally moved only in response to a comparison).

So (assuming that an assignment statement takes no more time than a comparison) we can multiply the above runtimes by two to get a rough upper estimate that comes closer to actual behavior in those cases that do in fact perform many moves. It happens that on average, Quicksort performs only about 1/3 as many moves as comparisons, while the others perform proportionately more.