Comparison sorts

Name	Best	Average	Worst	Memory	Stable	Method	Other notes
Quicksort	$n \log n$ variation is n	$n \log n$	n^2	$\log n$ on average, worst case space complexity is n; Sedgewick variation is $\log n$ worst case.	Typical in-place sort is not stable; stable versions exist.	Partitioning	Quicksort is usually done in-place with O(log n) stack space.[4][5]
Merge sort	$n \log n$	$n \log n$	$n \log n$	n A hybrid block merge sort is O(1) mem.	Yes	Merging	Highly parallelizable (up to O(log n) using the Three Hungarians' Algorithm[6] or, more practically, Cole's parallel merge sort) for processing large amounts of data.
In-place merge sort	—	—	$n \log^2 n$ See above, for hybrid, that is $n \log n$	1	Yes	Merging	Can be implemented as a stable sort based on stable in-place merging.[7]
Heapsort	$n \log n$	$n \log n$	$n \log n$	1	No	Selection	
Insertion sort	n	n^2	n^2	1	Yes	Insertion	O(n + d), in the worst case over sequences that have d inversions.
Selection sort	n^2	n^2	n^2	1	No	Selection	Stable with O(n) extra space, for example using lists.[8]
Shell sort	$n \log n$	$n \log^2 n$ or $n^{5/4}$	Depends on gap sequence; best known is $n \log^2 n$	1	No	Insertion	Small code size, no use of call stack, reasonably fast, useful where memory is at a premium such as embedded and older mainframe applications. Best case n log n and worst case n $\log^2$ n cannot be achieved together. With best case n log n, best worst case is $n^{4/3}$.
Bubble sort	n	n^2	n^2	1	Yes	Exchanging	Tiny code size.

The following table describes integer sorting algorithms and other sorting algorithms that are not comparison sorts. As such, they are not limited by a O(n log n) lower bound. Complexities below assume n items to be sorted, with keys of size k, digit size d, and r the range of numbers to be sorted. Many of them are based on the assumption that the key size is large enough that all entries have unique key values, and hence that $n \ll 2^k$, where $\ll$ means "much less than". In the unit-cost random access machine model, algorithms with