Characteristics of Sorting Algorithms

	Worst-Case Time	**Average-Case Time**	**Best-Case Time**	**Worst-Case Space**
BubbleSort ** more writes to memory than SelectionSort* ** adaptive* ** stable* ** in-place*	*$O(n^2)$ comparisons* *$O(n^2)$ swaps*	*$O(n^2)$ comparisons* *$O(n^2)$ swaps*	*$O(n)$ comparisons* *$O(1)$ swaps*	*$O(n)$ total* *$O(1)$ auxiliary*
InsertionSort ** more writes to memory than SelectionSort* ** adaptive* ** stable* ** in-place* ** online*	*$O(n^2)$ comparisons* *$O(n^2)$ swaps*	*$O(n^2)$ comparisons* *$O(n^2)$ swaps*	*$O(n)$ comparisons* *$O(1)$ swaps*	*$O(n)$ total* *$O(1)$ auxiliary*
SelectionSort ** double comparisons compared with InsertionSort* ** not stable but can be modified* ** not adaptive* ** in-place*	*$O(n^2)$ comparisons* *$O(n)$ swaps*	*$O(n^2)$ comparisons* *$O(n)$ swaps*	*$O(n^2)$ comparisons* *$O(1)$ swaps*	*$O(n)$ total* *$O(1)$ auxiliary*
HeapSort ** not stable but can be modified* ** not adaptive* ** in-place*	*$O(nlogn)$*	*$O(nlogn)$*	*$O(nlogn)$ distinct keys* *$O(n)$ equal keys*	*$O(n)$ total* *$O(1)$ auxiliary*

	Worst-Case Time	**Average-Case Time**	**Best-Case Time**	**Worst-Case Space**
MergeSort *Divide&Conquer* ** not adaptive* ** stable* ** not in-place in typical implementation (can be for linked lists)* ** accesses data sequentially and the need of random access is low* ** used for external sorting*	*O(nlogn)*	*O(nlogn)*	*O(nlogn)* *typical* *O(n)* *natural variant*	*O(n)* *total* *O(n)* *auxiliary with arrays* *O(1)* *auxiliary with linked lists*
QuickSort *Divide&Conquer* ** adaptive* ** not stable but can be modified (typically not implemented stable)* ** in-place* ** with better partition strategy (choosing pivot) worst-case time is rare !* ** needs random access*	$O(n^2)$ *comparisons* $O(n^2)$ *swaps*	*O(nlogn)*	*O(nlogn)* *simple partition* *O(n)* *3-way partition with equal keys*	*O(n)* *auxiliary (naive)* *O(logn)* *(Sedgewick 1978)*