

## Algoritmi di ordinamento

Tempi reali di esecuzione su due PC:

Test		field1 (sec)	field2 (sec)	field3 (sec)
InsertionSort	PC 1	> 10 min	> 10 min	> 10 min
	PC 2	> 10 min	> 10 min	> 10 min
MergeSort	PC 1	14.51	10.71	10.20
	PC 2	19.74	11.33	12.24
HeapSort	PC 1	58.39	58.39	58.46
	PC 2	55.87	45.80	55.35
QuickSort	PC 1	16.59	11.98	13.27
	PC 2	25.06	14.57	16.17

Complessità teorica degli algoritmi:

	Worst Case	Average Case	Best Case
Insertion Sort	$O(n^2)$	$O(n^2)$	$O(n)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick Sort	$O(n^2)$	$O(n \log n)$	$O(n \log n)$

INSERTION SORT: il tempo di esecuzione di insertion sort è superiore a 10 minuti, tale risultato era prevedibile in quanto l'algoritmo ha complessità quadratica sia nel caso medio che nel caso peggiore (che si verifica quando l'array è ordinato in ordine decrescente).

MERGE SORT: merge sort risulta molto performante, richiede però una struttura di appoggio che spreca spazio di memoria. Le sue prestazioni erano prevedibili in quanto, a differenza del quick sort, la sua complessità è  $O(n \log n)$  anche nel caso peggiore.

HEAP SORT: heap sort è affidabile e si comporta bene anche nel caso peggiore, risulta però generalmente più lento di quick e merge sort anche a parità di complessità.

QUICK SORT: quick sort risulta molto veloce per dati distribuiti in modo casuale. Il caso peggiore però ha complessità quadratica e si verifica quando le partizioni sono sbilanciate, ad esempio se si sceglie come pivot il primo elemento con un array ordinato in ordine crescente o decrescente.

Il caso peggiore si può limitare usando una scelta del pivot casuale.

Un risultato inaspettato durante i test si è verificato per l'ordinamento sul campo field1 (stringhe), in quanto nel file molti campi erano ripetuti e tale distribuzione provocava uno sbilanciamento delle partizioni. Abbiamo risolto il problema incrementando e decrementando contemporaneamente gli indici di inizio e fine della partizione nei campi uguali al pivot, in modo tale da rendere le partizioni più bilanciate.