

Relazione ex_1

Studente: Simone Milanese

Matricola: 946397



- Implementare una **libreria** che offre i due algoritmi di ordinamento **Merge Sort** e **Quick Sort**

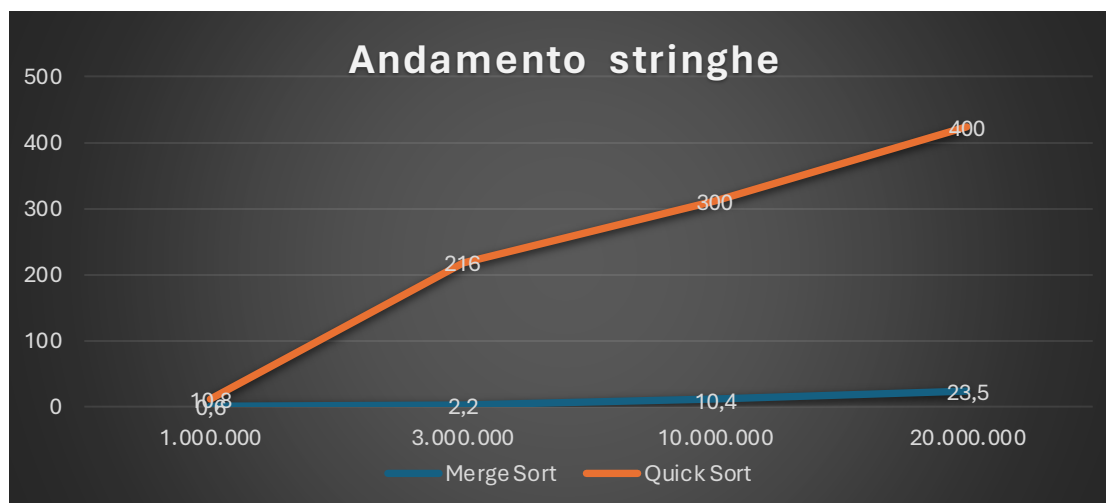
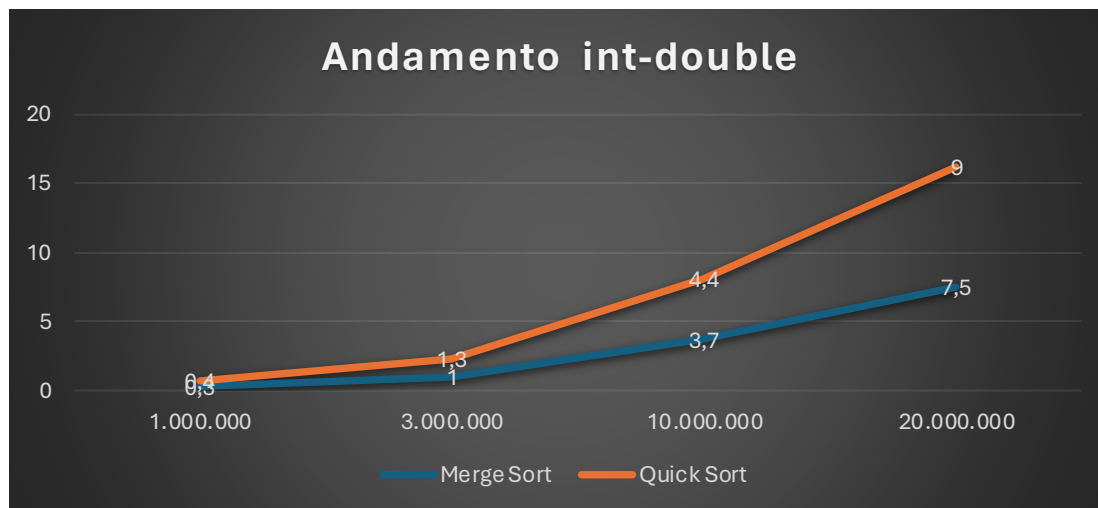
Scelte implementative

L'implementazione del Merge Sort presenta le due funzioni principali: `merge_sort()` e `merge()`. La prima ricorsiva, la seconda iterativa.

L'implementazione del Quick Sort presenta diverse versioni testate:

1. Quick Sort ricorsivo con partizione classica. Ultimo elemento come pivot.
2. Quick Sort iterativo.
3. Quick Sort 3-ways partition, che divide l'array in 3 parti, elementi minori del pivot, elementi uguali ed elementi maggiori.

Prestazioni 1 metodo



Considerazioni

I risultati ottenuti dai 2 algoritmi sono quelli attesi. I tempi di esecuzione del Merge Sort e del Quick Sort base risultano pressochè simili per quanto riguarda l'ordinamento di interi e double. Per quanto riguarda le stringhe, il tempo di ordinamento del Quick Sort base risulta notevolmente peggiore.

Dopo alcuni test di ordinamento sulle stringhe, notando che il dataset presentava un alto numero di ripetizioni ho modificato l'implementazione della funzione partition con quella del partizionamento a 3 vie.

I tempi di ordinamento di 20.000.000 di record del 3-ways Quick Sort sono stati:

- Int/Double: circa 15 secondi
- Stringhe: 22 secondi circa

I tempi di esecuzione dimostrano un tempo di ordinamento notevolmente ridotto rispetto alla precedente implementazione, ma solo per quanto riguarda i record di tipo stringa, a dimostrazione che questa scelta implementativa è ottima quando vi sono numerose ripetizioni all'interno dell'array da ordinare.