Relazione Primo Esercizio

Macchina usata per i test: MacBook Pro (Retina, 13-inch, Early 2015)

Ambiente di sviluppo: Eclipse IDE

Dataset: int.csv , sums.txt

Primo Utilizzo

Insertion Sort:tempo di esecuzione > 10 minuti. Programma interrotto.

Merge Sort:

Loading data from /Users/Silvestro/eclipse-workspace/int.csv
End

Loading data from /Users/Silvestro/eclipse-workspace/numbers.txt End

Sum 178045131 found

Sum 154266748 found

Sum 93573006 found

Sum 191011796 found

Sum 63744743 found

Sum 139183879 found

Sum 102116441 found

Sum 166015103 found

Sum 100000000 found

No sum 100 found

No sum 40000000000 found

No sum 1 found

No sum 2 found

No sum 3 found

No sum 4 found

No sum 54 found

No sum 57567 found

Sum 67863576 found

No sum 567567657676457 found

No sum 54 found

No sum 2 found

No sum 1 found

No sum 9 found

No sum 0 found

No sum 75 found

Total time 60 sec

Il programma è terminato in 60 secondi.

Conclusioni:
I due algorimti provati sullo stessa macchina, con lo stesso insieme di dati, hanno comportato tempi di esecuzione nettamente diversi.
Questo perchè Insertion Sort ha complessità O(n^2) mentre Merge Sort O(n * log(n)).

Secondo Utilizzo

Sfruttiamo il fatto di avere un array già ordinato.
Partiamo dai due estremi e verifichiamo che la loro somma sia uguale a sum, se non lo è decrementiamo o right o left. Si ripete per ogni numero del file.

Dalle prove effettuate l'algoritmo non sembra appesantire l'esecuzione e il tempo richiesto è di circa 10 secondi.